

Mall för planering och reflektion

Uppgift 3b, planering och reflektion

"Skapa en klass Fraction som representerar ett bråkital på formen T/N där T (täljaren) och N (nämnaren) är heltal. Om nämnaren är noll skall ett felmeddelande lämnas. Det skall finnas följande medlemmar i klassen.

- En konstruktor som skapar och initialiserar ett nytt bråkital.
- Metoderna `getNumerator` och `getDenominator` som returnerar täljaren respektive nämnaren.
- Metoden `isNegative` som ger `true` om det är ett negativt bråkital.
- Metoderna `add` och `multiply` som utför motsvarande bråktalsoperation på två bråk och som returnerar ett nytt bråkital. Bestäm själv ett lämpligt sätt att hantera de fall där någon av de inblandade bråktalen har noll i nämnaren.
- `isEqualTo` som jämför två Fraction-instanser och ser om de representerar samma bråkital.
- `toString` som returnerar en strängrepresentation av bråket på form T/N .

Lägg gärna till fler metoder om ni tycker att något fattas. Lämpliga argument- och returtyper för de olika metoderna får ni bestämma själva.

Överkurs för den intresserade: Se till så alla bråkital alltid är maximalt förenklade. T ex, bråktalen $2/4$ och $35/50$ skall internt representeras som $1/2$ och $7/10$. D v s, den interna representationen skall alltid vara de två minsta möjliga heltalen T och N som representerar samma rationella tal T/N . Här kan det vara nyttigt att titta på Euklides algoritm (se Wikipedia)."

Planering och tidsuppskattning

Planering och förarbete: 1 h 15 min,

varav gå igenom uppgiften: 0 h 15 min,

göra en skiss på struktur för programmet: 0 h 15 min,

läsa på om metoder för lösningar: 0 h 40,

och planera uppgiften: 0 h 5 min.

Genomförande: 2 h

varav skapa projektfiler: 0 h 5 min,

skriva källkoden och testa programmet i en iterativ process: 1 h 55 min.

Utvärdering och reflektion: 0 h 45 min

Sammanställa samtliga filer och kontrollera dessa (leverans): 0 h 15 min

Totalt: 4 h 15 min

Arbetet ska genomföras enligt principen att jag arbetar 50 minuter, för att sedan ta en paus på 10 minuter. Därför behöver den totala tid som avsätts varje dag ta med i beräkningarna att jag behöver mer tid avsatt än den tid jag faktiskt arbetar.

Utvärdering och reflektion

Den här uppgiften var klurig att planera vettigt. Kravspecifikationen är inte tydlig. Jag ska skriva en klass, men hur ska denna användas? Ska jag skriva ett program för att användaren ska kunna mata in bråk för att sen genomföra olika beräkningar, eller ska jag skriva ett program som bara testat detta?

Jag tog mig an uppgiften som så att en användare ska kunna mata in två bråk för att sedan genomföra olika beräkningar. Programmet i sig är jag inte nöjd med, det är ett rätt så konstigt program. Under tiden jag skrev programmet gick det nämligen mer och mer upp för mig att det snarare var ett test jag skulle skriva utöver klassen. Så programmet får ses som ett test där man kan mata in vilka bråk som helst - alltså programmet använder inte några förutbestämda testbråk. Detta gör att jag ändå kan acceptera att programmet i sig kan uppfattas lite konstigt, det ska helt enkelt bara användas för att testa klassen.

Att det blev mer och mer tydligt för mig under arbetet att jag missuppfattat uppgiften gjorde att jag tänkte om flera gånger, och till slut överskred jag tiden ganska kraftigt för genomförandet med 30 minuter. Jag hade 15 minuter över från planeringsfasen, vilket gjorde att tiden överskreds med 15 minuter totalt.

Att jag hade tid över från planeringsfasen kan tolkas som att jag inte planerade tillräckligt och därför fick jobba om min kod flera gånger i genomförandefasen - en bra planering och en tydlig kravspecifikation är alltså a och o.

Den struktur jag hade tänkt mig fungerade inte heller, och inte mycket finns kvar av den i slutändan.

I och med att frustrationen byggdes upp mer och mer tog jag mig inte heller tid för korta pauser, vilket antagligen hade hjälpt till för att rensa hjärnan och tänka om.

Så, vad lär jag mig av detta? Om kravspecifikationen är otydlig, fråga! Gissa inte. Vem vet, jag kanske fortfarande har uppfattat den fel, även nu när jag anser mig vara klar med ett färdigskrivet program.

Total uppskattad tid för uppgiften: 4 h 15 min

Total använd tid för uppgiften: 1 h 50 min

För mer information, se filen *Tidslog labb 1 - Iterativ Mjukvaruutveckling*, som finns i mappen *Övergripande dokument*.

Samtliga filer för denna uppgift finns på:

<https://github.com/lw222gu/1dv404-laboration1/tree/master/Uppgift%203/Uppgift%203b>