

Tekniskt PM

1- Introduktion

Precis som varje kurs eller varje programmering språk man använder, så ställer man ett krav på sig själv att man ska skriva så modulär kod som möjlig, trots att det är inte alltid detta som händer i slutändan. I denna kurs TDP028 med Java och Android är tanken likadan. Därför valde jag att ta fram om vad innebär modulär kod, hur uppnår man detta samt varför är det viktigt att använda.

2- Vad innebär modulär kod?

Moduläritet är en vanlig, känd och användbar metod när man arbetar med programmering även om de flesta som ingår med i programmerings samhälle inte faktiskt gör det eller inte uppnår kod moduläriteten, eftersom det är inte alltid enkelt att genomföra [1].

Att skriva modulär kod innebär att skriva ett program som är uppdelat i flera oberoende moduler. Därefter döljs implementerings detaljerna för dessa moduler, medan det publika gränssnittet utsatt endast för andra delar av koden. Detta gör det enklare att hantera sådana moduler, eftersom de kan testas, byggas om av sig själva eller till och med bytas ut mot en annan modul med samma publika gränssnittet om det behövs [2].

3- Hur uppnår man detta?

Alla programmerare kan uppnå modulär kod om de följer vissa instruktioner, en viktig instruktion som jag vill belysa som vi alltid bör tänka på när vi skriver kod eller skapar en viss modul/funktion är att den modulen ska genomföra exakt ett enskilt uppdrag som oftast beskrivs av modulens namn, detta betyder med andra ord att denna modul inte bör genomföra andra uppdrag eller andra uppgifter som inte beskrivs av modulens namn [1]. Detta gäller också för detta projekt som vi har i Android och Java där man exempelvis skapar en funktion som heter "darkMode" som bara gör en sak och detta är att byta mellan mörkt och ljus läge.

4- Varför är det viktigt?

Det finns säkert flera orsaker som visar oss varför ska vi skriva så modulär kod som möjligt, men det som är enkelt att se snabbt när man skriver modulär kod är att koden inte blir komplicerat [1]. De andra fördelar med detta är att man minskar antal misstag i programmet då blir det

lättare att undersöka eftersom man behöver undersöka endast den intresserande koden/modulen, det blir enkelt att förstå för de andra programmerare som behöver arbeta med eller läsa koden för att utveckla programmet, öka möjligheten att skapa olika tester för att kontrollera att koden genomför den uppgiften som den är implementerats till samt blir det enklare att återanvända dessa moduler/kod i framtiden var som helst i programmet [3].

Referenser:

- [1] D.Davis, J. Burry, and M. Burry, "Understanding Visual Scripts: Improving Collaboration through Modular Programming," *Int. J. Archit. Comput.*, vol. 9, no. 4, pp. 361–375, Dec. 2011, doi: 10.1260/1478-0771.9.4.361.
- [2] "2. Maintainability depends on modularity: Stop using namespaces!" <http://singlepageappbook.com/maintainability1.html> (accessed Dec. 28, 2021).
- [3] J. M. Trauer, R. Ragonnet, T. N. Doan, and E. S. McBryde, "Modular programming for tuberculosis control, the 'AuTuMN' platform," *BMC Infect. Dis.*, vol. 17, no. 1, Art. no. 1, Dec. 2017, doi: 10.1186/s12879-017-2648-6.