

# TDP028 Projekt: Entreprenöriell programmering

## Tekniskt PM

Albin Dahlén, [albda746@student.liu.se](mailto:albda746@student.liu.se)

## Innehåll

<b>1</b>	<b>Introduktion</b>	<b>2</b>
<b>2</b>	<b>Hur skapar man modulär kod</b>	<b>2</b>
<b>3</b>	<b>Varför använder man modulär kod</b>	<b>2</b>
<b>4</b>	<b>Modulär kod i android</b>	<b>2</b>
<b>5</b>	<b>Referenser</b>	<b>3</b>

## 1 Introduktion

Modulär kod är något som gör det enklare att strukturera och återanvända mycket kod. Då projektet jag arbetar på under kursen är stort och många sidor kommer ha liknande egenskaper med några få skillnader är det lämpligt att återanvända fragments och liknande för att inte behöva skriva samma kod flera gånger. För att komma fram till hur jag ska lösa detta behöver dessa frågor besvaras:

- Hur skapar man modulär kod?
- Varför använder man modulär kod?
- Hur skriver man modulär kod i Android?

## 2 Hur skapar man modulär kod

Vid skapandet av modulär kod är det viktigt att en modul har ett tydligt syfte och modulen ska bara utföra en sak medan den gömmer implementationen bakom ett väldefinierat gränssnitt. Modulerna ska även fungera självständigt och med andra moduler utan att behöva veta hur de andra modulerna är implementerade.<sup>1</sup> Exempelvis om en modul heter "displayPrograms" så ska modulen endast visa alla programmen för en användare och inget annat.

## 3 Varför använder man modulär kod

Genom att använda modulär kod går koden att återanvända genom att den är skriven på ett abstrakt sätt. Den modulära koden ska ha ett tydligt gränssnitt för hur den ska interageras med och detta leder till att utvecklarna för andra moduler inte behöver fundera på hur implementationen är de behöver bara använda gränssnittet när de ska koppla ihop modulerna.<sup>2</sup> Innan modulen implementeras i produkten kan den enskilt testas för att säkerställa att den fungerar som den ska. Genom att bryta ner koden i mindre moduler kan flera jobba på projektet samtidigt med en egen modul.<sup>3</sup>

## 4 Modulär kod i android

För att skriva modulär kod i android kan man använda sig av principen Model View Presenter. Denna principen säger att man delar in moduler i tre delar, en del har hand om logiken som att hämta data från en databas, en annan del har hand om hur detta visas för användaren medan den sista delen kopplar ihop dessa delar.<sup>4</sup> Exempel på detta skulle i android kunna vara att view-delen är ett fragment. Fragmentet tar in en lista av olika program som argument och det enda fragmentet sedan gör att att visa listan för användaren. Detta gör att vi lätt kan ta samma fragment och skicka använda i en annan activity där vi vill ha samma funktionalitet, vilket leder till att vi får modulär kod i android.

<sup>1</sup>Wasserman, A. I. (1996). Toward a discipline of software engineering *IEEE Software*. 13(6), s. 23-31. doi: 10.1109/52.542291.

<sup>2</sup>Wasserman, A. I. (1996). Toward a discipline of software engineering *IEEE Software*. 13(6), s. 23-31. doi: 10.1109/52.542291.

<sup>3</sup>Ross, D. T., Goodenough, J. B., och Irvine, C. A. (1975). Software Engineering: Process, Principles, and Goals. Computer. *IEEE Software* 8(5), s. 17-27. doi: 10.1109/C-M.1975.218952.

<sup>4</sup>Prabowo, G., Suryotrisongko, H., och Tjahyanto, A. (2018). A Tale of Two Development Approach: Empirical Study on The Maintainability and Modularity of Android Mobile Application with Anti-Pattern and Model-View-Presenter Design Pattern. *International Conference on Electrical Engineering and Informatics (ICELTICS) 2018, Banda Aceh, Indonesia*, s. 149-154. doi: 10.1109/ICELTICS.2018.8548784.

## 5 Referenser

Wasserman, A. I. (1996). Toward a discipline of software engineering. *IEEE Software*. 13(6), s. 23-31. doi: 10.1109/52.542291.

Prabowo, G., Suryotrisongko, H., och Tjahyanto, A. (2018). A Tale of Two Development Approach: Empirical Study on The Maintainability and Modularity of Android Mobile Application with Anti-Pattern and Model-View-Presenter Design Pattern. *International Conference on Electrical Engineering and Informatics (ICELTICS) 2018, Banda Aceh, Indonesia*, s. 149-154. doi: 10.1109/ICELTICS.2018.8548784.

Ross, D. T., Goodenough, J. B., och Irvine, C. A. (1975). Software Engineering: Process, Principles, and Goals. *Computer. IEEE Software* 8(5), s. 17-27. doi: 10.1109/C-M.1975.218952.