

Sammanfattning

I denna C-uppsats redogör vi för arbetet att revidera ett tidsrapporteringsprogram (THEO-tidsregistrering) ägt av IT-företaget Benefit Partner AB. Programmet skulle revideras därför att det inte alltid fungerade som det skulle och nya önskemål angående funktionalitet hade uppkommit.

Programmet är skrivet i Visual Basic och använder en Oracle-databas för informationslagring.

När vi skrev om programmet rensade vi först den ursprungliga koden från kodrader som inte längre användes, såsom gammal testkod. Under denna rensning fick vi även en första överblick av hur programmet fungerade. Därefter skrevs det mesta av koden om för att få ett stabilare, snabbare och bättre skrivet program.

Den främsta skillnaden i den nya versionen gentemot den tidigare är att funktionerna har blivit fler men generellare och mindre. Koden har optimerats så att ingenting längre körs i onödan och många inmatningstester och felkontroller har kunnat tas bort då inmatningen blivit mer restriktiv och felkällor har eliminerats.

Det problem vi främst stött på under arbetets gång har varit att den kod och databas vi hade som utgångsmaterial nästan helt saknade kommentarer och förklaringar.

Resultat är ett program som är snabbare, kortare och stabilare. Källkoden är lättläst, logisk och dokumenterad, vilket medför att programmet kommer vara mycket lättare att förstå för eventuell vidareutveckling.

THEO time-registration

Abstract

The purpose of this bachelor's project was to update a time-report system (THEO- time-registration) owned by the company Benefit Partner AB. The program needed an update because of unreliable execution and new ideas concerning the functionality.

The program is written in Visual Basic and uses an Oracle database to store information.

When we rewrote the program, we first cleared the original source code from parts that was no longer in use. This gave us a first overview of how the program worked. After that, most of the source code was rewritten to get a more stable, faster and an overall better program.

The foremost difference in the new version compared to the previous one is that the number of functions has increased but they are more generalized and not as big in size. The source code has been optimized so no code executes in vain and many input tests and error controls have been removed. This is due to more restrictive input and elimination of error sources.

The main problems we have encountered during this project have been due to lack of comments and explanations of the source code and the database.

The result is a program that is faster, shorter, more stable and nicer. The source code is readable, logic and well documented, which will make it easier to understand for further development.