Sammanfattning

Denna D-uppsats syftar till att beskriva bakgrunden till och tillkomsten av en utvecklingsmiljö för Java som har stöd för kontraktsprogrammering. Dokumentet specificerar vad syntax och semantik är, detta krävs för att man skall kunna sätta sig in i resten av dokumentet. Begreppet kontraktsprogrammering förklaras ingående då detta är ett nyckelbegrepp i utvecklingsmiljön och därmed denna uppsats. En genomgång av ett antal verktyg som används inom området kontraktsprogrammering görs och testas för att demonstrera deras funktionalitet. Dokumentet presenterar sedan ett idealsystem som vore det bästa möjliga resultatet av arbetet. Detta idealsystem används sedan för att jämföra det egentliga resultatet av arbetet med. Resultatet är en utvecklingsmiljö för Java som har stöd för att visa för- och eftervillkor vid metodanrop. Utvecklingsmiljön stöder även visning av klassinvarianter när nya instanser av klasser skall göras. Resultatet är alltså en utvecklingsmiljö som stöder kontraktsprogrammering. Denna utvecklingsmiljö ska förenkla för programmeraren att använda sig av kontraktsprogrammering vid utveckling av mjukvara. Författarnas förhoppning är att denna utvecklingsmiljö ska kunna bidra till bättre och mer pålitlig mjukvara.

Development Environment for Java with Support for Programming by Contract

Abstract

The purpose of this master's thesis is to describe the background to and creation of a development environment for Java that has support for programming by contract. The document specifies what syntax and semantics are since this is needed to understand the rest of the document. The conception of programming by contract is thoroughly explained since this is a key notion in the development environment and therewith also in this document. A review of some of the tools that is used within the area of programming by contract is made. These tools are also tested to demonstrate their functionality. The document then presents an ideal system that would be the best possible result of the project. This ideal system is later on used to compare the actual result of the project to. The result is a development environment for Java that has support for showing pre- and postconditions when invoking a method. The development environment also supports showing class invariants when new class instances are made. Consequently the result is a development environment that supports programming by contract. The intent of this development environment is to simplify for the programmer to use contracts in software development. It is the authors' hopes that this will lead to better and more reliable software.

Kommentar [TB1]: Var ska vi lägga in Tack-grejen? Efter Abstracten verkar det lite trångt…