Aufgabenblatt 8
SPFT

Tobias Petsch

Aufgabe 1

- **Vorteile:**
  - Programm kann direkt im Quelltext spezifiziert werden
  - Automatische Methoden zum Überprüfen des Codes können ausgeführt werden
  - einige Methoden funktionieren gut auf großen Programmen
  - praktische Implementierung von theoretischen Methoden zur Überprüfung von Programmen auf Korrektheit
  - Fehler können schnell erkannt werden
  - Code kann direkt kommentiert werden
- **Nachteile:**
  - Unit Tests können unter Umständen nicht alles überprüfen
  - Frage des Halteproblems kann auftreten
  - erfordert Ahnung über Spezifizierung
  - Asserts können die Performance des Programms einschränken

Design by Contract wird nicht umfänglich eingesetzt, weil nicht viele Sprachen dieses Tool unterstützen, zudem ist die Spezifizierung ein großer Aufwand bei Projekten die unter einer strikten Deadline stehen. Man muss auch abwiegen ob es sich lohnt, Performance einzubußen. Andere Methoden wie Unit Tests oder Test Driven Development haben sich stärker etabliert.

Aufgabe 2

- **Unit-Tests:**
- Generieren automatisch Unit tests für Funktionen und Klassen und decken dabei verschiedene Bereiche ab
  - **Vorteile:**
    - Automatische Testgenerierung
    - Schnelle Fehlerentdeckung bei Entwicklungprozess
  - **Nachteile:**

        * Mögliches fehlen von Abdeckung (Halteproblem)

- **Verifikation:**
- Programm wird formal verifiziert und beim Model Checking überpüft
  - **Vorteile:**
    * Analyse zur Designzeit
    * Keine false Alarme
  - **Nachteile:**
    * aufwendig
    * nur bei programme moderater Größe anwendbar
- **Dokumentation:**
- Generiert automatisch Dokumentation für Funktionen und Klassen in Einbezug der Bedingungen
  - **Vorteile:**
    * Schnelle Dokumentationserstellung
  - **Nachteile:**
    * Unter Umständen verwirrend formuliert
- **Runtime Assertion Testing:**
- Generiert im Code Assertions hinein welche bei Ausführung des Codes Fehler werfen können
  - **Vorteile:**
    * Code wird durch asserts modifiziert
    * Eigenes Verhalten zur Fehlerverwaltung implementierbar
  - **Nachteile:**
    * Fehler erst bei Laufzeit bekannt
    * teuer in Performance
- **Statische Analyse:**
- Übersetzt Code in SMT kompatibles Format wo es dann durch einen SMT Solver gelöst werden kann
  - **Vorteile:**
    * Überprüfung der Korrektheit des Programms
    * sehr große Programme möglich
    * Überprüfung zur Designzeit möglich
  - **Nachteile:**
    * Mögliche False Positives und false negatives