|  |  |
| --- | --- |
| **Programmierschnittstellen und Softwarequalität**  **Prof. Dr. Ursula Oesing** |  |
| **Projekttermin 4: Modultest mit JUnit**  **und Äquivalenzklassentest** |

**4. Modultests mit JUnit**

**4.1 Informationen und Voraussetzungen zu diesem Projekttermin**

Vorhanden ist die Lösung des dritten Projekttermins. Es werden jetzt Modul-tests mit JUnit für den Client des Projekts durchgeführt.

**4.2 Aufgaben dieses Projekttermins**

**4.2.1 Modultest mit JUnit mit Parametern**

Erstellen Sie einen Modultest mittels JUnit, welcher den Konstruktor der Klasse *Messreihe* mit dem folgenden Testfall testet: Die MessreihenID sei *1*, das Zeitintervall *20*, der Verbraucher *LED* und die Messgröße *Leistung.* Erweitern Sie die Testmethode unter Verwendung eines Parameters, so dass auch der Testfall (1, 20, LED, Arbeit) getestet wird.

**4.2.2 Erstellen von Testfällen nach dem Äquivalenzklassentest**

Überlegen Sie sich weitere Testfälle nach dem Äquivalenzklassentest unter Berücksichtigung ungültiger Werte und Verschmelzung, siehe unten Schrei-ben Sie die Äquivalenzklassen auf, Sie erhalten 5 oder 6 Klassen. Erstellen Sie für jede Äquivalenzklasse einen Testfall. Die Definitionsmengen der Parameter sind folgende.

messreihenId Menge aller int-Werte

zeitintervall {15, 16, 17, …}

verbraucher Menge aller Strings \ {““, null}

messgroesse {“Arbeit“, “Leistung“}

Verschmelzen Sie zwei Äquivalenzklassen, falls diese beide für denselben  
Parameter einen ungültigen Wert haben. Als Repräsentanten nehmen Sie dann einen, der für keinen zweiten Parameter einen ungültigen Wert hat.

**4.2.3 JUnit Tests mit Parametern inklusive Testen von Exceptions**

Erstellen Sie zu einem Testfall aus 4.2.2 mit ungültigen Werten einen JUnit Test unter Verwendung von Exceptions. Korrigieren Sie den zu testenden Konstruktor, falls der Testfall nicht erfolgreich durchläuft.