

OTH Regensburg
Fakultät für Informatik und Mathematik
Beispielklausur MST/WMST
bei Prof. Dr. M. Bulenda

Name, Vorname:

Matrikelnummer:.....

Hinweise:

- Schreiben Sie nicht mit Bleistift, rotem oder grünem Stift.
- Verwenden Sie für die Lösungen den dafür vorgesehenen Platz direkt bei den jeweiligen Aufgaben.
- Schreiben Sie leserlich.
- Sollten Sie ein oder mehrere weitere Blätter abgeben wollen, so heften Sie die Blätter zur Klausur. Keine lose eingelegten Blätter!
- Die Klausur umfasst 12 Seiten. Überprüfen Sie, ob Ihnen alle Seiten vorliegen.

Mit Ihrer Unterschrift bestätigen Sie, dass Sie die Hinweise gelesen haben.

Unterschrift:

Teil	Erreichbare Punkte	Erreichte Punkte	Note
Teil 1 - Theorie	58		
Teil 2 - Praxis	42		
Summe	100		

Unterschrift Erstprüfer:

Ggfs. Unterschrift Zweitprüfer:

Teil 1 – Theorie

Aufgabe 1 Multiple Choice

Punkte: ____

10 Punkte

Kreuzen Sie an, ob die jeweilige Aussage richtig oder falsch ist. Jede richtige Antwort gibt einen Punkt, jede falsche Antwort einen Minuspunkt. Nicht angekreuzte Fragen geben 0 Punkte. Insgesamt mindestens 0 Punkte.

Id	Frage	Richtig	Falsch
1	Wenn Sie in einem Projekt eine hinreichend hohe Testabdeckung haben, sind statische Codeanalysen nicht mehr notwendig.		
2	Sie können das Qualitätsmerkmal „Benutzbarkeit“ eines Systems optimieren ohne die anderen Qualitätsmerkmale nach ISO/IEC 9126-1 zu beeinträchtigen, da keine Korrelation zu den anderen Merkmalen besteht.		
3	Das Schichtenmodell zur Bearbeitung von Konflikten von Berg besagt, dass die Bearbeitung von Konflikten immer an der tiefsten Ebene, also der Ebene der Persönlichkeit, beginnen soll.		
4	Ein Vorteil von toolgestützten Codereviews ist, dass der für den gereviewten Code verantwortliche Entwickler sich nicht durch den Reviewer persönlich angegriffen fühlt.		
5	Bei der Verwendung von Code Metriken können Sie entweder Metriken der imperativen Programmierung verwenden oder Metriken der Objektorientierung, aber nicht beide gemeinsam.		
6	Software Metriken werden meist manuell erstellt.		
7	In wichtigen Softwareprojekten sollten Sie immer eine 100%ige Pfadüberdeckung bei Ihren Tests anstreben.		
8	Das Tool gcov liefert keine Aussage zur Anweisungsüberdeckung, sondern nur zur Zeilenüberdeckung.		
9	Tools zur statischen Code Analyse werden unter anderem dazu verwendet, um die Einhaltung projektspezifischer Code Konventionen zu überprüfen.		
10	Scrum ist nicht für große Projekte geeignet.		

Aufgabe 2 Kommunikation

Punkte: ____

12 Punkte

Beschreiben Sie das Kommunikationsmodell nach Friedemann Schulz von Thun.

Aufgabe 3 Software Qualität Qualitätssicherung

Punkte: ____

6 Punkte

Nennen Sie zwei typische Notationskonventionen für Programm Code und warum es jeweils sinnvoll ist, diese Konvention zu nutzen.

Aufgabe 4 Software Qualität

Punkte: ____

2 Punkte

Erklären Sie den Unterschied zwischen statischer und dynamischer Typprüfung.

Aufgabe 5 Software Qualität

Punkte: ____

8 Punkte

Erklären Sie die **Grundprinzipien** des Prinzips Design By Contract.

Aufgabe 6 Software Qualität

Punkte: ____

3 Punkte

Nennen Sie drei Vorteile der Praxis, Kommentare im Code zu pflegen und daraus zu extrahieren.

Aufgabe 7 Versionsverwaltung

Punkte: ____

4 Punkte

Sie haben in einem neuen Projekt die Wahl zwischen zwei Versionsverwaltungssystemen: GIT und Subversionen.

Nennen Sie jeweils einen Vorteil und einen Nachteil jedes Systems gegenüber dem anderen.

GIT:

Vorteil gegenüber Subversion:

Nachteil gegenüber Subversion:

Subversion:

Vorteil gegenüber GIT:

Nachteil gegenüber GIT:

Aufgabe 8 Maven

Punkte: ____

4 Punkte

Sie sind in Ihrer Firma für ein Software Modul (in Form eines jar Files) verantwortlich, das von vielen internen Projekten genutzt wird. Innerhalb Ihrer Firma verwenden sowohl Sie als auch die Projekte, die Ihr Modul nutzen, Maven zur Build Automatisierung. Ihr Modul soll nicht von außerhalb der Firma verwendbar sein.

Durch welches Tool ermöglichen Sie es den nutzenden Projekten, Ihr jar File zuzugreifen und welche Maven-Koordinaten nutzen die zugreifenden Systeme, um Ihr jar File zu finden?

Antwort:

Tool:

Koordinaten:

Aufgabe 9 Agile Methoden

Punkte: ____

4 Punkte

Agile Methoden setzen sich immer mehr durch. Dennoch gibt es auch kritische Stimmen. Nennen Sie zwei Gründe, warum agiles Vorgehen mit Problemen verknüpft sein kann. (Geben Sie nicht nur Stichpunkte an, sondern formulieren Sie die Probleme verständlich aus).

Antwort:

Aufgabe 10 JUnit

Punkte: ____

5 Punkte

Erklären Sie, wofür JUnit Rules verwendet werden. Nennen Sie ein konkretes Beispiel, wo die Verwendung von JUnit Rules sinnvoll ist.

Antwort:

Teil 2 Praxis

Aufgabe 1 Mc Cabe Metrik

- a) Wie ist die McCabe Metrik für einen zusammenhängenden Graphen definiert?
2 Punkte

- b) Erstellen Sie für folgende Funktion den Kontrollflussgraphen:
6 Punkte

Kontrollflussgraph

```
int find_maximum(int *input_array, int length){  
  
    int max_value;  
    int index_of_max_value;  
    int index;  
    if(length <= 0){  
        printf("Please enter valid values.\n");  
        return -1;  
    }  
    max_value = input_array[0];  
    index_of_max_value = 0;  
    for(index=1; index < length; index++){  
        if(input_array[index] > max_value){  
            max_value = input_array[index];  
            index_of_max_value = index;  
        }  
    }  
    return index_of_max_value;  
}
```

- c) Berechnen Sie für den gerade erstellten Graphen den Wert der McCabe Metrik:
2 Punkte

Aufgabe 2 Tests

8 Punkte

Betrachten Sie den Code aus Aufgabe 1. Definieren Sie für die folgenden Aufgaben Testfälle in der Form:

Testfall_<Nr>: input_array = {<Kommaseparierte Liste der Elemente>}, length = <Länge>

- a. Definieren Sie eine möglichst geringe Anzahl von Testfällen, so dass Sie eine 100%ige Anweisungsüberdeckung erhalten.

- b. Definieren Sie eine möglichst geringe Anzahl von Testfällen, so dass Sie eine 100%ige Zweigüberdeckung erhalten.

- c. Wie viele Testfälle erhalten Sie für eine komplette Pfadüberdeckung?
Begründen Sie Ihre Antwort.

Aufgabe 3 Datenflussanomalien

12 Punkte

Sie finden in einem Java Programm folgenden Code für eine Funktion

Int swap(int i, int j, int[] arr)

vor. Die Funktion erhält zwei primitive int Variablen i und j und den Zugriff auf ein int Array als Funktionsparameter. Aufgabe der Funktion ist es, die Elemente des Arrays an den Positionen i und j zu vertauschen, sofern das Element an der Stelle i größer ist als das Element an der Stelle j.

Erstellen Sie einen Kontrollflussgraphen, eine Datenflussanalyse und kennzeichnen Sie ggfs. Stellen, die auf einen potentiellen Fehler hinweisen.

Hinweis: Die Elemente des Arrays werden mit Verlassen der Funktion NICHT ungültig.

Funktion

```
int swap(int i, int j, int[] arr) {  
    int temp;  
    int ret=0;  
    if( arr[i]>arr[j] ){  
        temp = arr[i];  
        arr[j] = arr[i];  
        arr[i] = temp;  
        ret=1;  
    }else{  
        ret=0;  
    }  
    return ret;  
}
```

Kontrollflussgraph

Anomalienanalyse

Aufgabe 4 Maven

12 Punkte

Stellen Sie eine Maven Installation für ein Software Projekt mit einem Architekturbild dar. Erklären Sie damit das Konzept der lokalen und remote Repositories und das Plugin Konzept von Maven.

Antwort:

Bild:

umblättern!

Erklärung Konzept der lokalen und remote repositories

Erklärung Plugin Konzept