

Probe-Klausur Software Engineering

Fachbereich BW, für WINFO

Dipl.-Ing. Klaus Knopper

17.04.2007

Hinweis: Bitte schreiben Sie auf das Deckblatt und auf jede Seite Ihren Namen und Ihre Matrikelnummer, bevor Sie mit der Bearbeitung der Aufgaben beginnen!

Kontrollieren Sie bitte, ob Sie alle 5 Aufgabenblätter vollständig erhalten haben.

Die Aufgaben lassen sich alle direkt auf dem jeweiligen Aufgabenblatt lösen lassen. Wenn Sie dennoch mehr Platz benötigen, können Sie die Rückseite der Aufgabenblätter unter Angabe der Aufgabennummer benutzen.

Viel Erfolg!

Name: _____ Matrikelnummer: _____

1 Allgemeines

1.1 Was versteht man unter der „Softwarekrise“? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|---|---|
| <input type="radio"/> Mangel an Software | <input type="radio"/> Mangel an Qualität der Software |
| <input type="radio"/> Preisverfall der Software | <input type="radio"/> Versions-Chaos |

1.2 Erklären Sie in jeweils einem kurzen Satz die Begriffe „Verifikation“ und „Validierung“ in Bezug auf Software-Produkte. (2 Punkte)

1.3 Was wird durch das „Lastenheft“ festgelegt? (4 Punkte)

(Bitte bei jeder Antwort ein Kreuz in das richtige Feld einfügen. Richtig angekreuzte Antworten ergeben einen Punkt. Falsch angekreuzte Antworten führen zu einem Punkt Abzug. Nicht angekreuzte Antworten werden nicht bewertet. In der Gesamtwertung trägt die Aufgabe keine negativen Punkte bei.)

	Trifft zu	Trifft nicht zu
Katalog von Anforderungen	<input type="radio"/>	<input type="radio"/>
Das Gewicht der Software	<input type="radio"/>	<input type="radio"/>
Die technischen Details	<input type="radio"/>	<input type="radio"/>
Verwendete Entwurfsmuster	<input type="radio"/>	<input type="radio"/>

1.4 Nennen Sie zwei Hauptaufgaben des Projektmanagers? (2 Punkte)

1.5 Welchen Nachteil hat das Vorgehensmodell „Extreme Programming“ gegenüber Modellen mit umfangreicherer Planungsphase? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|--|---|
| <input type="radio"/> Programmierer werden verheizt | <input type="radio"/> Scheitern vorprogrammiert |
| <input type="radio"/> Strategische Fehler werden oft erst spät erkannt | <input type="radio"/> Versions-Chaos |

2 Lizenzen

2.1 Nennen Sie 4 essentielle Freiheiten, die „Open Source“-Lizenzen definitionsgemäß den Empfängern der Software geben. (4 Punkte)

2.2 Welches der folgenden Rechte bleibt immer beim Urheber, und kann nicht veräußert oder gemindert werden? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|--|---|
| <input type="radio"/> Verwertungsrecht | <input type="radio"/> kommerzielles Nutzungsrecht |
| <input type="radio"/> Urheberrecht | <input type="radio"/> Recht zur kostenlosen Verbreitung |

3 Planung & Analyse

3.1 Die Anforderungsanalyse... (2 Punkte)

(Bitte bei jeder Antwort ein Kreuz in das richtige Feld einfügen. Richtig angekreuzte Antworten ergeben einen Punkt. Falsch angekreuzte Antworten führen zu einem Punkt Abzug. Nicht angekreuzte Antworten werden nicht bewertet. In der Gesamtwertung trägt die Aufgabe keine negativen Punkte bei.)

	Trifft zu	Trifft nicht zu
wird im direkten Dialog mit den Auftraggebern/Kunden durchgeführt	<input type="radio"/>	<input type="radio"/>
ist ein wesentlicher Bestandteil des SE, um Qualität und Produktivität zu sichern	<input type="radio"/>	<input type="radio"/>
wird ausschließlich mit mathematischen und statistischen Methoden durchgeführt	<input type="radio"/>	<input type="radio"/>
legt fest, welche Entwurfsmuster verwendet werden	<input type="radio"/>	<input type="radio"/>

3.2 Welcher der hier genannten Punkte gehört NICHT zu den Kern-Phasen der Anforderungsanalyse? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|--|---|
| <input type="radio"/> Aufnahme und Erfassung | <input type="radio"/> Strukturierung und Abstimmung |
| <input type="radio"/> Prüfung und Bewertung | <input type="radio"/> Evaluation eines Prototypen |

4 Unified Modelling Language

...die wichtigsten UML-Diagrammtypen erkennen, oder angeben, welcher Diagrammtyp zweckmäßig ist, auffällige Fehler oder Unklarheiten in UML-Diagrammen finden.

5 Entwurf

5.1 Wofür steht der Begriff CASE bezüglich Software Engineering? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|--|--|
| <input type="radio"/> Eine bedingte Verzweigung in JAVA | <input type="radio"/> Eine Programmiersprache |
| <input type="radio"/> Computerunterstütztes Software Engineering | <input type="radio"/> CAD-Programm zur Erstellung von UML-Diagrammen |

5.2 Was ist ein „evolutionärer Prototyp“? (2 Punkte)

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- | | |
|--|--|
| <input type="radio"/> Ein veralteter BASIC-Dialekt | <input type="radio"/> Ein Programmierfehler |
| <input type="radio"/> Eine kontinuierlich erweiterte „Testversion“ | <input type="radio"/> Künstliche Intelligenz |

5.3 Welche Vorteile bieten interfaces gegenüber classes in Java bei der Softwareentwicklung? (4 Punkte)

(Bitte bei jeder Antwort ein Kreuz in das richtige Feld einfügen. Richtig angekreuzte Antworten ergeben einen Punkt. Falsch angekreuzte Antworten führen zu einem Punkt Abzug. Nicht angekreuzte Antworten werden nicht bewertet. In der Gesamtwertung trägt die Aufgabe keine negativen Punkte bei.)

	Trifft zu	Trifft nicht zu
Sie sind schneller	<input type="radio"/>	<input type="radio"/>
Sie kosten weniger Speicher	<input type="radio"/>	<input type="radio"/>
Sie erlauben trotz fehlender Mehrfachvererbung in Java die Vererbung mehrerer „Klassen von Methoden“	<input type="radio"/>	<input type="radio"/>
Sie sorgen dafür, dass der Programmierer auch alle für eine Schnittstelle benötigten Methoden implementiert	<input type="radio"/>	<input type="radio"/>

6 Entwurfsmuster in Java

6.1 Welches Entwurfsmuster wird hier verwendet? (2 Punkte)

```
import java.util.Observable;
import java.util.Observer;
public class Messung extends Observable {
    private float messwert = 0.0F; // Initialwert
    public void messung() { // Neuen Messwert holen
        float neu = Eingabe.readFloat(); // Messung simulieren
        if ( neu != messwert ) {
            messwert = neu;
            setChanged(); // Aus Observable
        }
        notifyObservers();
    }
    public float getMesswert() { return messwert; }
}
```

6.2 Welches der in der Vorlesung behandelten Entwurfsmuster passt am besten auf ein Problem, bei dem eine große Anzahl gleichartiger und nicht-unterscheidbarer Objekte performant verwaltet werden muss? (2 Punkte)

7 Tests

7.1 Finden Sie den Fehler in folgendem Programm, und kreisen Sie ihn ein. (2 Punkte)

```
public class Main {
    public static final int counter = 100;
    public static void main(String[] args) {
        if(counter-- > 0) {
            System.out.println("counter = " + counter);
            main(args); // Rekursion
        }
    }
}
```

7.2 Was soll mit einem Integrationstest untersucht werden? (2 Punkte)

Bei einem Integrationstest soll untersucht werden, ob...

(Zutreffendes bitte ankreuzen. Nur eine Antwort ist richtig.)

- alle Schnittstellen gemäß ihrer Spezifikation fehlerfrei arbeiten.
 alle Schnittstellen bei beliebigen Eingabewerten fehlerfrei arbeiten.
- alle Konstruktormethoden implementiert sind.
 sich das Integral $\int_{-\infty}^{\infty} p(x_i) dx_i$ berechnen lässt.

8 Begleitende Prozesse

8.1 Oft wird gesagt, der „Quelltext ist die beste Dokumentation“. Inwiefern ist dies nach Software-Engineering-Maßstäben zutreffend oder nicht? (4 Punkte)

Der Quelltext eines Programms ist...

(Bitte bei jeder Antwort ein Kreuz in das richtige Feld einfügen. Richtig angekreuzte Antworten ergeben einen Punkt. Falsch angekreuzte Antworten führen zu einem Punkt Abzug. Nicht angekreuzte Antworten werden nicht bewertet. In der Gesamtwertung trägt die Aufgabe keine negativen Punkte bei.)

	Trifft zu	Trifft nicht zu
allenfalls eine technische <u>System</u> dokumentation, keine Dokumentation für Endbenutzer.	<input type="radio"/>	<input type="radio"/>
nur als Dokumentation brauchbar, wenn er selbst gut strukturiert und dokumentiert ist.	<input type="radio"/>	<input type="radio"/>
bei Whitebox-Tests eine unerlässliche Komponente, um auch Fehlerquellen dokumentieren zu können.	<input type="radio"/>	<input type="radio"/>
besser als jedes Handbuch.	<input type="radio"/>	<input type="radio"/>
die häufigste Fehlerursache.	<input type="radio"/>	<input type="radio"/>