

Aufgabe

Erstellen Sie arbeitsteilig ein Softwaresystem und dokumentieren sie alle durchlaufenen Phasen des Software-Life-Cycles.

Das Softwaresystem soll auf dem in der LWB verwendeten Linux-System lauffähig sein und aus mindestens zwei Klassen/ADTs bestehen.

Arbeiten Sie in Gruppen von mindestens 2 bis maximal 4 Mitgliedern.

Anforderungen

Folgende Leistungen sind zu erbringen.

- Das zu erstellende Softwaresystem soll hinsichtlich der funktionalen und nichtfunktionalen Anforderungen analysiert werden. Die Anforderungen sollen im Lastenheft festgehalten werden. Sie können die beiliegende Vorlage dafür verwenden.
- Ein objektorientierter Entwurf des Softwaresystems als UML-Diagramm unter Verwendung des Diagrammeditors DIA als .dia-Datei und als pdf. Die pdf-Datei enthält eine kurze Darstellung der Entwurfentscheidungen.
- Eine genaue Spezifikationen der selbst erstellten Klassen in Dateien mit der Nomenklatur <Klassenname>.go.
- Eine arbeitsteilige Implementierung von Systemkomponenten in Dateien mit der Nomenklatur <Klassenname>Impl.go.
Dabei muss eine namentliche Zuordnung von Teilen der Implementierung zu den jeweiligen Gruppenmitgliedern erfolgen. Halten Sie dies im Quellcode als Kommentar fest und fertigen Sie eine stichwortartige tabellarische Übersicht über die jeweiligen Arbeitsanteile an. Die jeweiligen Anteile müssen einen vergleichbaren Umfang haben.
- Zu einzelnen Klassen ist ein Testprogramm vorhanden, welches die Funktionalitäten der Klasse sinnvoll überprüft. Nomenklatur <Klassenname>Test.go.
- Die Implementierung soll die Hinweise für gute Implementierung erfüllen und nachvollziehbar kommentiert sein.
- Gegebenenfalls ist eine Anleitung für die Verwendung des Softwaresystems beigelegt, wenn dies notwendig sein sollte.

Einzureichende Dokumente und Dateien

Am Tag der Abgabe sind folgende Dokumente und Dateien einzureichen:

- Eine Datei <Projektname>.tgz oder <Projektname>.zip mit folgenden Aufbau.

```
|
| – 01_Analyse.pdf
| – 02_Entwurf.dia
| – 02_Entwurf.pdf
| – 03_Projektuebersicht.pdf
| – 04_Implementierung
|   | – <Klasse1>
|   |   | – <Klasse1>.go
|   |   | – <Klasse1>Impl.go
|   | – <Klasse2>
|   |   | – <Klasse2>.go
|   |   | – <Klasse2>Impl.go
|   ...
|   | – <Klasse1>Test.go
|   | – <Klasse2>Test.go
|   | – <Projektname>.go
| – 05_Anleitung.pdf
```

- Die Analyse, das Klassendiagramm, die Spezifikationen der Klassen und die Projektübersicht in gedruckter Form, versehen mit einem Deckblatt (Name der Gruppenmitglieder, Bezeichnung des Softwareprojektes). Sollte bis zum Abgabetermin keine Präsenzveranstaltung stattfinden, so kann darauf verzichtet werden.

Bewertung

Die Bewertung erfolgt nach dem folgenden Bewertungsraster.

Software / Implementierung						
Kriterium	++	+	o	-	--	Bemerkungen
Das System funktioniert konfliktfrei. Alle benötigten Klassen sind vorhanden.						
Alle im Lastenheft festgelegten Funktionalitäten sind vorhanden.						
Die Implementierung entspricht der Spezifikation.						
Die Hinweise für eine gut Implementierung sind größtenteils eingehalten. Der Code ist nachvollziehbar kommentiert.						
Bezeichner von Klassen, Methoden und Variablen sind durchgängig sprechend gewählt.						
Die Implementierung ist durch passende Kommentierung im Quellcode nachvollziehbar.						
Die Architektur des Softwaresystems entspricht wichtigen Grundregeln der Softwaretechnik.						
Das System ergibt eine für den Nutzer verständliche, durchgängige, flüssig bedienbare Anwendung. (z. B. Startbildschirm, Hilfen/Spielregeln, Endbildschirm, neue Runde, Abbruchmöglichkeit)						
Es wurden über absolut notwendige Grundfunktionen hinausgehende Funktionen implementiert.						
Die Komplexität des Softwaresystems ist der Anzahl der Gruppenmitglieder und der Zeit angemessen.						
Für einzelne Klassen ist ein sinnvolles Testprogramm vorhanden,						
Dokumentation						
Es sind alle geforderten Dokumente und Dateien im geforderten Format termingerecht eingereicht worden.						
Die Analyse beschreibt alle geforderten Punkte ausreichend detailliert.						

Das Klassendiagramm enthält alle Implementierten Klassen, ist in der Notation korrekt und ausreichend präzise.						
Die Spezifikationen sind sprachlich sauber formuliert.						