

Aufgabenstellung zum Vorprojekt

1. Zielbestimmung

Das Ziel des Vorprojektes ist eine Einteilung des Gesamtprojektes in kleinere Teilprojekte, in denen sich Stück für Stück an das Gesamtprojekt vorgearbeitet wird. Dabei geht es hauptsächlich darum, in den einzelnen Teilprojekten Prototypen zu erstellen, welche alle für sich lauffähig sind und zusammen funktionieren können.

Diese Strukturierung dient in erster Linie dazu, die Aufgaben zu verteilen, eine gemeinsame Arbeitsstruktur zu entwickeln und dabei gesetzte Deadlines einhalten zu können.

Auch ermöglicht diese Einteilung für uns, dass jeder seine spezifischen Kenntnisse besser einsetzen kann und leichter neue Kenntnisse erworben werden können.

2. Projekteinteilung

Das Projekt werden wir in folgende Teile gliedern:

2.1 Erstellung des Codegenerators

2.2 Entwicklung einer browserbasierten Benutzeroberfläche für den Zugriff auf die relevanten Daten

2.3 Erstellung der konsistenten relationellen Datenbank, in welcher später die Lebenslaufakten und Objekte gespeichert werden können

2.4 Erstellung von Abfrageroutinen auf die Datenbank, durch welche die Lebenslaufakten gesucht, erstellt, bearbeitet und gegebenenfalls gelöscht werden können.

2.5 Tests auf Funktionsfähigkeit der Teilprojekte und gegebenenfalls erste Tests mit Zusammengeführten Projektteilen. Letzteres betrifft insbesondere die Datenbank mit den Abfrageroutinen.

3. Produktfunktionen

/PF21/ Der Codegenerator soll im Stande sein, nach UML2-Standard erstellte, UML-Modelle zu lesen und diese in einen kompilierbaren Quelltext umzuwandeln

/PF22/ Die Benutzeroberfläche soll in Java-Quelltext eingebettet sein und folgende Funktionen enthalten:

- Auswahl "bestehende Objekte einsehen", "bestehende Objekte bearbeiten", "neues Objekt erstellen"

- Suchansicht mit Texteingabe und Auswahl nach welchen Kriterien gesucht werden soll-->welches Objekt, welches Attribut
- Ansicht auf eine vollständige Lebenslaufakte mit allen Daten, die das gesuchte Objekt beinhaltet.
- + für Bearbeitungsmodus änderbare Eingabefelder und Button für Änderung speichern

/PF23/ funktionierende Datenbank mit vollständigen Relationen, welche sich an dem vorgegebenen UML-Modell orientiert

/PF24/ Routinen für den Datenbankzugriff, die folgendes beinhalten:

- Suchroutine mit spezifischer Suche nach vorhandenen Dateien
- Ablaufroutine, welche vorhandene Dateien ändert oder löscht. Dabei ist besonders darauf zu achten, dass die Abhängigkeiten der Klassen beachtet werden um Datenmüll zu vermeiden(Bsp.: ein gelöschter Park kann keine Anlagen mehr enthalten
- Abfragemaske zur Einbindung neuer Objekte und korrekte Eintragung in die vorhandene Datenbank

/PF25/ Testklassen, die jedes Programm auf seine Funktionsfähigkeit überprüfen

Datenbank		Qualität		Webseite
löschen	fügen	Formatierung	große Quelltextstruktur	Benutzeroberfläche mit Funktionalität
		Funktionalität	Kompilierbarkeitskriterien	