

3 Übungen Tests in bestehendem Code hinzufügen

3.1 Dateiüberprüfung

Wir haben einen kleinen Dateiüberprüfungstool übernommen.

Leider haben die Kunden damit Probleme, wenn sie '/' im Pfad verwenden.

Sie sollen den Fehler beheben und danach den Code auf den neusten Stand bringen. Schreiben Sie zuerst Tests die den Fehler aufzeigen und beheben Sie ihn erst dann. Testen Sie das gesamte Tool bevor Sie den Code modernisieren.

Tipps

Nutze "Extrahiere Schnittstelle" für die Kapselung der Readeraufrufe.

Nutze "Extrahiere in virtuelle Methode", das Erzeugen des konkreten Readers austauschen zu können.

Verwende `std::filesystem` und `std::filesystem::path` um den Code zu modernisieren.

Man kann `\\dd-samba\sys\test\TestSourceMaterial\` oder `\\localhost\freigabe` für den remote Test verwenden.

3.2 Namensverzeichnis

Unser Namensverzeichnis ist eine kleine Klasse, mit der wir neue Elemente hinzufügen und auslesen können. Außerdem können wir uns einen Index generieren lassen, der uns eine Übersicht über alle bisher hinzugefügten Elemente gibt. Schlecht ist, dass wir die Indexgenerierung mehrmals anstoßen können und wir dann ein falsches Ergebnis erhalten. Daher soll die Generierung des Index weniger fehleranfällig gestaltet werden, indem wir bei jedem Hinzufügen eines Elements, den Index aktualisieren. Allerdings soll das erst geschehen, wenn einmal die Indexgenerierung angestoßen wurde.

Zeigen Sie anhand einer Effektskizze, wo Änderungen gemacht und wo der Test am besten die Änderungen abprüfen sollte.

Mit dem Ausbilder die Lösung besprechen und dann weiter mit:

Das Verhalten dementsprechend ändern. Das Programm in der main.cpp darf nicht geändert werden.

3.3 Export Engine

Der PO wünscht sich, dass unser Export auch Graustufenvideos exportieren kann.

Refaktorisieren Sie die Exportfunktion `ExportEngine::Bounce` nach IOSP und PoMO, sodass die einzelnen Aspekte des Exports sichtbar werden.

Fügen Sie anschließend das Feature hinzu und testen Sie es automatisiert.

Tipps

Liste zuerst alle Aspekte in der `ExportEngine` auf und extrahiere jeden Aspekt in ein Modul, damit mögliche Tests erleichtert werden.

Aus dem `Videoframe` ein Graustufenbild zu machen, ist eine Art der Farbkonvertierung

(<https://de.wikipedia.org/wiki/Grauwert>)

3.4 Bezahlssystem

Das Bezahlssystem ist erst einige Tage im Einsatz und es gibt bereits massive Kritik von Arbeitern die nicht korrekt bezahlt werden. Das Management möchte das teure System aber nicht gleich wieder aufgeben und beauftragt Sie die Probleme zu beheben und dafür zu Sorgen, dass es nie wieder passiert.

Das System soll folgende Features bieten:

- Man kann Arbeiter mit verschiedenen Gehaltsarten dem System hinzufügen
 - Es gibt Arbeiter die werden stundenweise bezahlt. Sie geben täglich an, wieviel Stunden sie gearbeitet haben (via Zeitkarten). Überstunden (mehr als 8) werden mit dem 1.5-fachen Stundenlohn bezahlt. Die Auszahlung erfolgt immer am Freitag. Es ist möglich das es mehrere Zeitkarten für einen Tag gibt, dann müssen die Stunden addiert werden.
 - Es gibt Angestellte mit festem Monatslohn. Die Auszahlung erfolgt am letzten Tag des Monats.
- Mitarbeiter aus Deutschland haben eine spezielle Kostenberechnung (Kranken+Rentenversicherung) und daher muss diese änderbar sein
- Das System läuft einmal pro Tag und gibt die Lohnscheine an diesem Tag aus

Gemeldete Problem:

- Die Auszahlungen erfolgen nicht an dem richtigen Tag (z.B. 5.7.2019 ist Freitag, wird aber nicht ausgezahlt)
- Die Zeitspanne für den Lohnschein ist manchmal falsch (z.B. bei Jahreswechsel)
- Es wird nur die zuletzt abgegebene Zeitkarte an einem Tag gezahlt
- Das System gibt auch einen Lohnschein aus, wenn der Arbeiter in dem Zeitraum gar nicht gearbeitet hat

Als motivierter Entwickler stürzen wir uns in das System und beheben die Fehler mit Leichtigkeit.

Als professioneller Entwickler:

- Verschaffen uns zuerst mittels Skizze einen groben Überblick über das System
- Stellen sicher, dass das System nach jeder Änderung weiterhin so funktioniert wie gewollt
- Sorgen mit automatisierten Tests dafür, dass die Fehler nicht wieder passieren
- Machen nur die nötigen Änderungen an dem System, damit wir schnell zum Ziel kommen

Mit dem Ausbilder die Lösung der bisherigen Übung besprechen und dann weiter mit:

Das Management möchte Arbeiter mit Provision auch mit diesem System bezahlen lassen. Der Arbeiter sammelt die Quittungen und aus einem Prozentsatz der Summe der Erlöse ergibt sich die Provision. Diese wird einem Grundgehalt hinzugefügt. Die Auszahlung erfolgt am Freitag alle zwei Wochen, also insgesamt 26 mal im Jahr bei 52 Wochen im Jahr. Es ist möglich mehrere Quittungen an einem Tag einzureichen, dann muss der Verkaufserlös pro Tag addiert werden.