Semesterprojekt sysVentory

* Computerinformationen auf bestimmte Merkmale inventarisieren
* Abweichungen zwischen einzelnen Inventarisierungen visualisieren
* Daten müssen in einer Datei gespeichert werden
* Alle Versionen der einzelnen Inventarisierung müssen in dieser Datei hinterlegt sein
* Programm soll ab einem externen Datenträger (z.B. USB Stick) gestartet werden können, um unterschiedliche Rechner inventarisieren zu können

Hilfsmittel

* eine dll, welche den Scan durchführt und eine Liste mit allen definierten Items liefert. Dabei finden sich in der Liste Hard- wie auch Softwareinformationen

Funktionale Anforderungen

* Invetarisierungsscan kann manuell ausgelöst werden
* Daten werden in einer Datei strukturiert gespeichert, für jedes Gerät eine eigene Datei
* Scan wird mit Datum und Uhrzeit in Datei abgelegt
* Unterschiede zwischen zwei Scans können sichtbar visualisiert werden
* Bereits erfolgte Scans können jederzeit wieder betrachtet werden
* Die Inventarisierungs-Historie eines Elements kann betrachtet und die ganze Chronik hinsichtlich Scans und Details visualisiert werden

Optionale Anforderungen

* Alle Scan-Daten eines Elements können gelöscht werden
* Unterschiede von Hard- und Softwareinventare verschiedener Rechner können visualisiert werden

Nicht funktionale Anforderungen

* Applikation wird in VisualStudio 2019 mit C# und Windows-Forms erstellt
* Nur Form-Komponenten, welche im Unterricht besprochen und eingesetzt wurden
* Inventarkatalog wird visuell sinnvoll dargestellt (Hardware bzw. Software getrennt)
* Die Applikation hat eine einfache, aber durchdachte systematische Struktur (Zuständigkeiten von Klassen, Methoden usw. sind klar geregelt)
* Die Architektur der Applikation setzt das MVC-Pattern um
* Es werden die Konzepte der objektorientierten Programmierung umgesetzt
* Im VisualStudio ist ein Klassendiagramm erstellt und jederzeit aktuell zu halten

Bewertung

* Die erwartete Erfüllung eines Kriteriums wird mit der Note 5.0 bewertet
* Implementierungen, welche die Erwartung eines Kriteriums übertreffen, können höher bis maximal Note 6.0 bewertet werden

Bewertungsraster

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Kriterium** | 6.0 | 5.5 | 5.0 | 4.5 | 4.0 | 3.5 | 3.0 | 2.5 | 2.0 | 1.5 | 1.0 |
| Funktionale Anforderungen |  |  |  |  |  |  |  |  |  |  |  |
| Architektur der Applikation |  |  |  |  |  |  |  |  |  |  |  |
| Qualität der Benennung von Klassen, Methoden, Properties und Feldern |  |  |  |  |  |  |  |  |  |  |  |
| Qualität der Menge von Code in Klassen, Methoden sowie deren funktionale Zuständigkeit bzw. Abgrenzung |  |  |  |  |  |  |  |  |  |  |  |
| Benutzerfreundlichkeit hinsichtlich Bedienung |  |  |  |  |  |  |  |  |  |  |  |
| Qualität der Inline-Dokumentation |  |  |  |  |  |  |  |  |  |  |  |
| Stabilität der Applikation, Umgang mit Fehlern |  |  |  |  |  |  |  |  |  |  |  |

Abgabe

* Abgabe bis spätestens Montag, 14.09.2020, 23:00 Uhr, spätere Abgaben werden nicht berücksichtigt
* Die Arbeit wird als ZIP-Datei via Lernplattform eingereicht
* Innerhalb der ZIP-Datei befindet sich eine team.txt Datei, in welcher alle Teilnehmer mit Vorname und Nachname aufgeführt sind (Notenabzug von 0.5 falls nicht vorhanden)
* Komplettes C#-Projekt befindet sich innerhalb des ZIP-Ordners

Funktionale Anforderungen

* Hard- und Softwareinventarisierung
* Scan kann manuell gestartet werden
* Scan-Resultat wird strukturiert in einer Datei abgelegt
* Jedes Gerät/Element hat eine eigene Datei
* Alle Versionen der Scans eines Elements müssen in der Datei ersichtlich sein
* Historie der Scans eines Elements soll abrufbar sein
* Unterschiede zwischen den Scans eines Elements sollen visuell dargestellt werden
* Programm soll ab externem Datenträgen (z.B. USB Stick) auf verschiedenen Geräten gestartet werden können
* Unterschiede zwischen zwei Geräten (Soft- und Hardware) sollen visuell dargestellt werden **(optionale Funktionalität)**
* Alle Scans eines Elements sollen gelöscht werden können **(optionale Funktionalität)**

Nicht funktionale Anforderungen

* Applikation wird in VisualStudio 2019 mit C# und Windows-Forms erstellt
* Nur Form-Komponenten, welche im Unterricht besprochen und eingesetzt wurden
* Inventarkatalog wird visuell sinnvoll dargestellt (Hardware bzw. Software getrennt)
* Die Applikation hat eine einfache, aber durchdachte systematische Struktur (Zuständigkeiten von Klassen, Methoden usw. sind klar geregelt)
* Die Architektur der Applikation setzt das MVC-Pattern um
* Es werden die Konzepte der objektorientierten Programmierung umgesetzt
* Im VisualStudio ist ein Klassendiagramm erstellt und jederzeit aktuell zu halten

Abgabe

* bis spätestens Montag, 14.09.2020, 23:00 Uhr
* ZIP-Datei mit C# Projekt und team.txt Datei