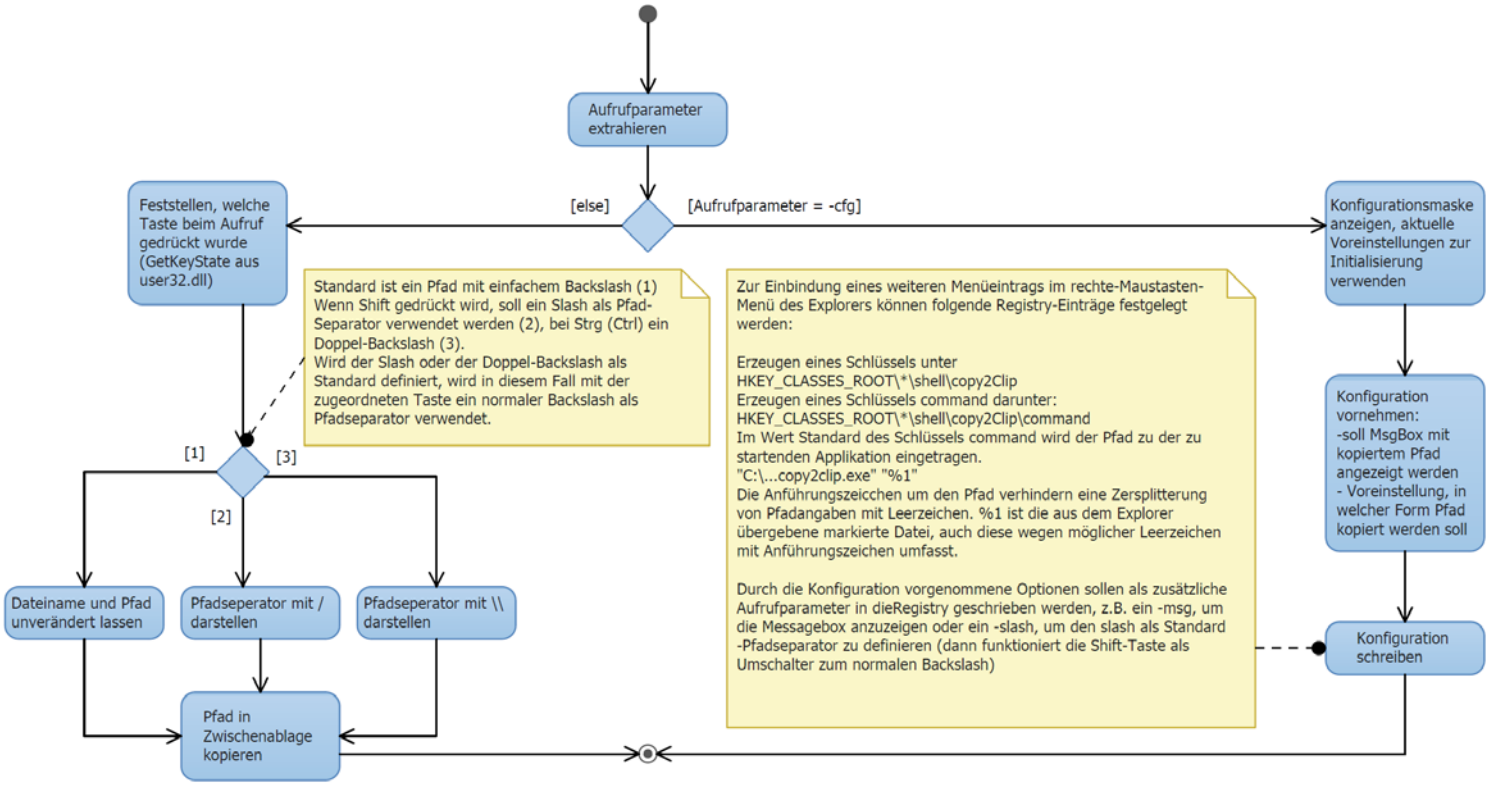
Prof. Dr.-Ing. Jörg Schlingheider
Studiengang Ingenieurinformatik
Wilhelminenhofstr. 75A
12459 Berlin
Tel.: 030-5019-4354
Email: joerg.schlingheider@htw-berlin.de

Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences
www.htw-berlin.de

Übung

Erstellen Sie eine Anwendung, mit der Sie über einen neuen Befehl im rechte-Maus-Tasten-Kontext-Menü des Explorers den Pfad und Dateinamen der aktuell selektierten Datei in die Windows-Zwischenablage kopieren können. Durch gleichzeitiges Drücken von Shift bzw. Strg (Ctrl) soll es möglich sein, den Pfadseparator auf Slash (/) oder Doppel-Backslash (\\) umzustellen.

Außerdem soll es möglich sein, in einer GUI kleinere Einstellungen zum Programmverhalten zu konfigurieren. Den Ablauf des Programms zeigt folgendes Aktivitäten-Diagramm:



Bevor Sie sich der Umsetzung widmen, klären Sie bitte folgende Fragen:

- Wie können Aufrufparameter in einem C#-Programm ermittelt werden
- Wie können Werte aus der Windows-Registry gelesen und geschrieben werden
- Wie kann die Windows-Zwischenablage mit Daten gefüllt werden

Die oben im Diagramm genannte Funktion GetKeyState kann prüfen, ob die entsprechende Taste gedrückt wurde, unabhängig von einem Steuerelement. Auch bei Steuerelementen kann dies u.U. erforderlich sein, da z.B. in manchen Fällen die

TAB-Taste kein Ereignis auslöst, da mit ihr zum nächsten Steuerelement gewechselt wird.

Die Funktion GetKeyState befindet sich in einer „normalen“ Windows-Bibliothek (user32.dll). Dies ist weder eine .NET- noch eine COM-Dll, die über einen Verweis eingebunden werden kann, sondern eine Standard-Dll, die im Gegensatz zu den genannten Dll-Typen keine unmittelbar verwendbaren Informationen bzgl. der enthaltenen Funktionen mitbringt.

Um derartige Dll trotzdem nutzen und aufrufen zu können, gibt es in C# die Klasse DllImportAttribute, bei der mit dem Statement

```
[DllImport("user32.dll", CharSet = CharSet.Auto, ExactSpelling = true)]
```

die user32.dll in den Adressraum geladen wird. Zusätzlich muß definiert werden, wie eine Funktion aus der Dll aufgerufen werden soll. Dazu ist die Funktion über ein weiteres Statement in C# bekannt zu machen:

```
/*private/public*/ static extern short GetKeyState(int virtualKeyCode);
```

Die importierte Funktion kann in der Klasse, in der der Import erfolgt, als public, private oder auch internal (nur innerhalb des Namensraums) deklariert werden, die Funktion selbst ist immer der Klasse zugeordnet (static) und als extern (also nicht dem aktuellen Projekt zugehörig) gekennzeichnet.

Wenn nun alle direkt erkennbaren Umsetzungsprobleme verstanden sind, versuchen Sie, wesentliche Klassen auf Basis des Aktivitäten-Diagramms als Basis der Umsetzung zu benennen. Dabei sollten Sie im Sinne der Wiederverwendung eine hohe Modularisierung anstreben.

Erstellen Sie nun eine Windows-Forms-Anwendung, legen die Klassen an und beginnen mit der Umsetzung.