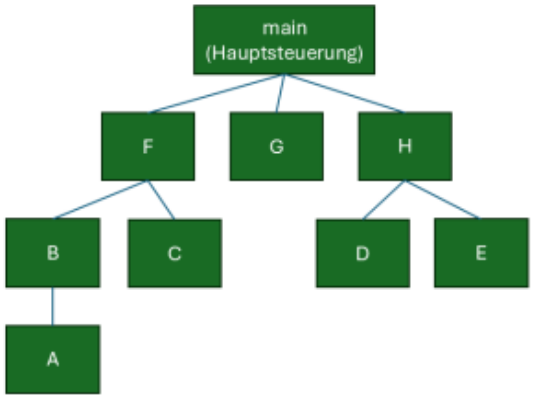
LU15b - Umsetzung von Integrationstests

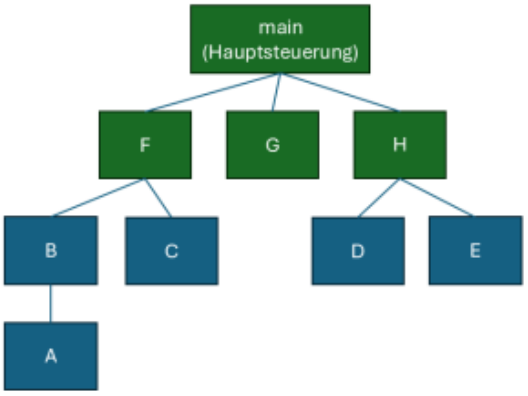
Im Gegensatz zu Unittests wollen wir bei Integrationstests das Zusammenspiel verschiedener Komponenten testen. Daher verzichten wir weitgehend auf simulierte Funktionen und Klassen (Mocks).

Big-Bang-Integration



Bei dieser Strategie wird das Zusammenspiel aller Komponenten (Module, Funktionen, Klassen) gleichzeitig getestet. Dabei verzichten wir vollständig auf Mocks. Anhand der Hauptsteuerung suchen wir die Hauptkomponenten unserer Applikation. Zu diesen Komponenten schreiben wir Integrationstests. Dabei werden alle Komponenten, die von den Hauptkomponenten angesprochen werden, automatisch auch getestet.

Top-Down-Integration

	Im Gegensatz zur Big-Bang-Integration testen wir immer nur eine Schicht auf einmal. Wir starten mit der höchsten Ebene der Anwendung, z. B. der Hauptsteuerung oder den GUI-Komponenten, und arbeiten uns schrittweise zu den unteren Schichten (z. B. Datenbankzugriff oder Services) vor. Mocks werden hier häufig eingesetzt, um noch nicht integrierte Unterschichten zu simulieren.

Erste Schicht

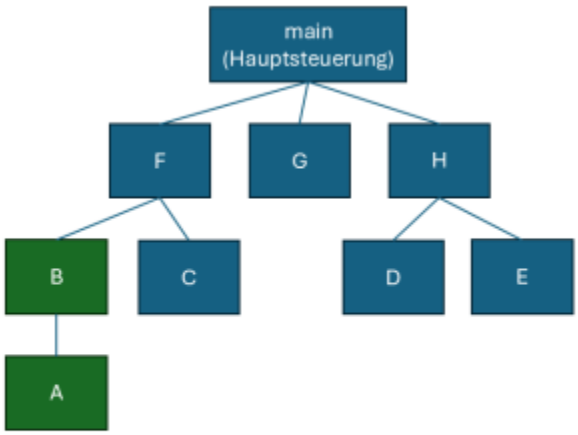
1. Wähle eine Komponente (nennen wir sie **function_F**) aus, die von der Hauptsteuerung aufgerufen wird.
2. Erstelle eine Testfunktion bei der die Hauptsteuerung die **function_F** aufruft.
3. Erstelle simulierte Funktionen für **B** und **C** (Mocks).
4. Führe die Tests aus: Dokumentiere dabei auftretende Fehler und behebe diese iterativ.

Wiederhole diese Schritte, bis die erste Schicht **F, G, H** vollständig getestet ist.

Weitere Schichten

In weiteren Testfällen prüfen wir nun die Komponenten, welche von der Komponente **F** aufgerufen werden. Dies wird nun Komponente für Komponente und Schicht für Sicht wiederholt.

Bottom-Up-Integration



Bei dieser Strategie testen wir zuerst die Integration von Komponenten, die keine weiteren Abhängigkeiten haben. Wir beginnen mit den grundlegenden Bausteinen des Systems (z. B. Datenbank- oder Utility-Funktionen) und arbeiten uns schrittweise nach oben. Höhere Schichten werden erst integriert, wenn ihre Abhängigkeiten bereits getestet wurden.

Basierend auf Unittests

Diese Strategie ist relativ einfach umzusetzen, wenn schon weitgehend vollständige Unittests existieren.

1. Identifiziere eine Testfunktion (nennen wir sie **test_A**), die keine Mocks einsetzt.
    - Diese Testfunktion wird in der Regel nur eine Funktion (**function_A**) testen.
    - Dies sind zu Beginn vor allem Tests von Komponenten, die keine weiteren Komponenten aufrufen.
2. Suche eine Testfunktion (**test_B**), welche einen Mock dieser Funktion **function_A** nutzt.
3. Erstelle eine Kopie von **test_B**, wobei du nun die echte **function_A** aufrufst.
4. Führe die Tests aus: Dokumentiere dabei auftretende Fehler und behebe diese iterativ.

Diese Schritte werden wiederholt, bis das Zusammenspiel aller Komponenten getestet wird.

M450-LU15