

Blatt 10

Software Engineering 2

Luis Staudt

Aufgabe 1

Testfall für User Story 1

User Story 1:

Als Dozent möchte ich Aufgaben mit allen notwendigen Attributen und Kategorisierungen in der Datenbank speichern können, damit ich eine umfangreiche Sammlung an Klausuraufgaben erstellen kann.

Testfall 1.1: Vollständige Aufgabe erfolgreich speichern

1. Anmeldung im System als Dozent mit gültigen Anmelddaten
2. Navigation zur Funktion "Neue Aufgabe erstellen"
3. Eingabe aller erforderlichen Attribute:
 - Name: "Relationale Algebra Grundlagen"
 - Aufgabentext: "Erklären Sie die grundlegenden Operatoren der relationalen Algebra und geben Sie je ein Beispiel an."
 - Modulzugehörigkeit: "Datenbanken I"
 - Geschätzte Bearbeitungszeit: "15 Minuten"
 - Bloom'sche Taxonomie: "Level 2 - Verstehen"
 - Aufgabenformat: "Offen"
 - Musterlösung: "Die grundlegenden Operatoren sind Selektion, Projektion, Vereinigung, Differenz, Kartesisches Produkt und Verbund..."
4. Bestätigung der Eingabe durch Klick auf "Speichern"
5. Überprüfung der Erfolgsmeldung: "Aufgabe erfolgreich gespeichert"
6. Verifikation in der Aufgabenübersicht: Die gespeicherte Aufgabe erscheint in der Liste mit allen eingegebenen Attributen
7. Filtertest: Aufgabe kann über alle definierten Filterkriterien gefunden werden

Erwartetes Ergebnis:

Die Aufgabe wird vollständig mit allen Attributen in der Datenbank gespeichert und ist über die Suchfunktionen auffindbar.

Testfall für User Story 2

User Story 2:

Als Dozent möchte ich Klausuren aus vorhandenen Aufgaben nach definierten Kriterien zusammenstellen können, damit ich effizient Prüfungen erstellen kann.

Testfall 2.1: Klausur nach Filterkriterien zusammenstellen

1. Anmeldung im System als Dozent
2. Vorbereitung: Mindestens 5 verschiedene Aufgaben mit unterschiedlichen Attributen sind in der Datenbank vorhanden
3. Navigation zur Funktion "Klausur zusammenstellen"
4. Anwendung der Filterkriterien:
 - Modul: "Datenbanken I"
 - Bloom'sche Taxonomie: Level 1-3 (Grundlagen bis Anwendung)
 - Format: Gemischt (offen und geschlossen)
5. Überprüfung der Filterergebnisse: Nur Aufgaben, die den Kriterien entsprechen, werden angezeigt
6. Auswahl von 4 Aufgaben durch Markierung
7. Überprüfung der automatischen Zeitberechnung: Gesamtzeit wird korrekt summiert angezeigt
8. Speicherung der Klausurzusammenstellung unter dem Namen "Datenbanken I - Zwischenprüfung"
9. Bestätigung der Speicherung durch Erfolgsmeldung
10. Verifikation: Klausur erscheint in der Übersicht gespeicherter Klausuren
11. Exporttest: Klausur kann als PDF-Datei exportiert werden

Erwartetes Ergebnis:

Eine Klausur wird erfolgreich aus gefilterten Aufgaben zusammengestellt, die Gesamtzeit wird korrekt berechnet, und die Klausur kann gespeichert und exportiert werden.

Aufgabe 2

Für die Kaffeemaschine werden folgende Unit-Test-Klassen erstellt, um alle wichtigen Methoden und Zustandsübergänge abzudecken:

Testabdeckung

Zu testende Klassen und Methoden:

1. **CoffeeMachine Klasse:**

- Konstruktor und Initialisierung
- setState() - Zustandswechsel
- Getter/Setter für Zutaten und Münzguthaben
- hasEnoughIngredients() - Zutatenprüfung

2. **State-Klassen:** Alle Zustandsübergänge für jeden State

- IdleState
- CoinAcceptedState
- CheckingIngredientsState
- PreparingCoffeeState
- CoffeeReadyState
- ErrorState

Testszenarien:

- **Normale Abläufe:** Vollständiger Kaffeezubereitungsprozess
- **Fehlerfälle:** Fehlende Zutaten, mehrfache Münzeinwürfe
- **Grenzfälle:** Zustandsübergänge bei verschiedenen Bedingungen
- **Zutatenverwaltung:** Auffüllen und Verbrauch von Ressourcen

Aufgabe 3

Identifikation der Äquivalenzklassen

Für die Bestellmenge mit dem gültigen Bereich 1–1000 Stück ergeben sich folgende Äquivalenzklassen:

Klasse	Beschreibung	Bereich	Erwartetes Verhalten
ÄK1	Ungültig (zu klein)	< 1	Ablehnung
ÄK2	Gültig	$1 \leq x \leq 1000$	Annahme
ÄK3	Ungültig (zu groß)	> 1000	Ablehnung

Tabelle 1: Äquivalenzklassen für Bestellmenge

Grenzwertanalyse

Die kritischen Grenzwerte sind:

Grenzwert	Wert	Erwartetes Verhalten
Untere Grenze - 1	0	Ablehnung
Untere Grenze	1	Annahme
Untere Grenze + 1	2	Annahme
Obere Grenze - 1	999	Annahme
Obere Grenze	1000	Annahme
Obere Grenze + 1	1001	Ablehnung

Tabelle 2: Grenzwertanalyse für Bestellmenge

Testfälle

Test-ID	Beschreibung	Eingabe	Erwartetes Ergebnis	Äquivalenzklasse
T1	Negative Bestellmenge	-5	Fehlermeldung: "Bestellung abgelehnt"	ÄK1
T2	Bestellmenge Null	0	Fehlermeldung: "Bestellung abgelehnt"	ÄK1
T3	Minimum gültige Menge	1	Bestellung akzeptiert	ÄK2
T4	Menge knapp über Minimum	2	Bestellung akzeptiert	ÄK2
T5	Mittlere gültige Menge	500	Bestellung akzeptiert	ÄK2
T6	Menge knapp unter Maximum	999	Bestellung akzeptiert	ÄK2
T7	Maximum gültige Menge	1000	Bestellung akzeptiert	ÄK2
T8	Menge über Maximum	1001	Fehlermeldung: "Bestellung abgelehnt"	ÄK3
T9	Sehr große Menge	5000	Fehlermeldung: "Bestellung abgelehnt"	ÄK3

Tabelle 3: Testfälle für Online-Bestellung

Aufgabe 4

Identifikation der Äquivalenzklassen

Für ein Programm zur Dreieckerkennung mit drei ganzzahligen positiven Werten ergeben sich folgende Äquivalenzklassen:

Klasse	Beschreibung	Erwartetes Ergebnis
ÄK1	Ungültige Eingabe (nicht-positive Werte)	Fehlermeldung
ÄK2	Keine Dreiecksungleichung erfüllt	Fehlermeldung
ÄK3	Gleichseitiges Dreieck ($a = b = c$)	"Gleichseitiges Dreieck"
ÄK4	Gleichschenkliges Dreieck (zwei Seiten gleich)	"Gleichschenkliges Dreieck"
ÄK5	Ungleichseitiges Dreieck (alle Seiten verschieden)	"Ungleichseitiges Dreieck"

Tabelle 4: Äquivalenzklassen für Dreieckerkennung

Detaillierte Äquivalenzklassen

Mathematische Bedingungen:

- **Gültige Dreiecke:** $a + b > c$ und $a + c > b$ und $b + c > a$
- **Gleichseitig:** $a = b = c$
- **Gleichschenklig:** $a = b \neq c$ oder $a = c \neq b$ oder $b = c \neq a$
- **Ungleichseitig:** $a \neq b \neq c \neq a$

Testfälle

Test-ID	Beschreibung	Eingabe (a,b,c)	Erwartetes Ergebnis	ÄK
T1	Negative Werte	(-1, 5, 3)	Fehlernmeldung	ÄK1
T2	Null-Werte	(0, 4, 5)	Fehlernmeldung	ÄK1
T3	Dreiecksungleichung verletzt (1)	(1, 2, 5)	Fehlernmeldung	ÄK2
T4	Dreiecksungleichung verletzt (2)	(1, 1, 3)	Fehlernmeldung	ÄK2
T5	Gleichseitiges Dreieck	(5, 5, 5)	“Gleichseitiges Dreieck”	ÄK3
T6	Gleichseitiges Dreieck (klein)	(1, 1, 1)	“Gleichseitiges Dreieck”	ÄK3
T7	Gleichschenklig ($a=b$)	(5, 5, 3)	“Gleichschenkliges Dreieck”	ÄK4
T8	Gleichschenklig ($a=c$)	(4, 6, 4)	“Gleichschenkliges Dreieck”	ÄK4
T9	Gleichschenklig ($b=c$)	(3, 5, 5)	“Gleichschenkliges Dreieck”	ÄK4
T10	Ungleichseitig	(3, 4, 5)	“Ungleichseitiges Dreieck”	ÄK5
T11	Ungleichseitig (groß)	(13, 14, 15)	“Ungleichseitiges Dreieck”	ÄK5
T12	Grenzfall Dreiecksungleichung	(1, 2, 3)	Fehlernmeldung	ÄK2

Tabelle 5: Testfälle für Dreieckerkennung

Zusätzliche Grenzfälle:

- **Minimale gültige Werte:** (1, 1, 1) - Gleichseitig
- **Grenzfall Gleichschenklig:** (2, 2, 3) - Gerade noch gültiges gleichschenkliges Dreieck
- **Grenzfall Ungleichung:** (5, 5, 9) - Gerade noch gültiges gleichschenkliges Dreieck
- **Große Werte:** (100, 150, 200) - Test mit größeren Zahlen