Contents

[1. JUnit 2](#_Toc181547110)

[1. **TestInMemoryDatabaseRetrieval** 2](#_Toc181547111)

[2. **TestReceiptProcessing**: 2](#_Toc181547112)

[3. **TestDonationFunctionality** 3](#_Toc181547113)

[4. T 3](#_Toc181547114)

[2. Definitionen zur rechtlichen Verbindlichkeit 3](#_Toc181547115)

[1. MUSS 3](#_Toc181547116)

[**2.** SOLLTE 4](#_Toc181547117)

[3. WIRD 4](#_Toc181547118)

[3. Funktionale Anforderungen 5](#_Toc181547119)

[**1.** Prüfung und Annahme des Artikels 5](#_Toc181547120)

[**2.** Anzeige und Benutzerinteraktion 5](#_Toc181547121)

[**3.** Sortierung und Lagerung von Artikeln 5](#_Toc181547122)

[**4.** Pfandbetragsberechnung und -ausgabe 6](#_Toc181547123)

[**5.** Benachrichtigungen und Rückmeldungen 6](#_Toc181547124)

[4. Konzept des Pfandautomaten (PA) 7](#_Toc181547125)

[Schritt 0: Das System benennen 7](#_Toc181547126)

[Schritt 1: Die rechtliche Verbindlichkeit festlegenn 8](#_Toc181547127)

[Schritt 2: Funktionalität identifizieren 8](#_Toc181547128)

[Schritt 3: Art der Funktionalität 9](#_Toc181547129)

[Schritt 4: Das Objekt identifizieren 9](#_Toc181547130)

[Schritt 5: Die Bedingungen formulieren 10](#_Toc181547131)

## 1. JUnit

**Definition:** Zuerst müssen wir die Klassen und Methoden importieren und deklarieren, die wir brauchen.

@BeforeEach  
void setUp() {  
 display = new Display();  
 cardScanner = new CardScanner(bottleDepositMachine);  
 bottleDepositMachine = new BottleDepositMachine(display);  
 supervisoryModule = new SupervisoryModule(bottleDepositMachine, cardScanner, display);  
 inMemoryDatabase = new InMemoryDatabase();  
 receiptProcessor = new ReceiptProcessor();  
 donationDatabase = new DonationDatabase();  
}

**Aufgaben des Junit-Tests:**

1. **TestInMemoryDatabaseRetrieval:** Wir prüfen, ob ein Artikel mit dem Barcode „r8yz7clkz4“ in inMemoryDatabase existiert.  
    assertNotNull(item)  
    Wenn ja, dann ist das Frontlabel gleich „ABC Can | 0.33L“  
   assertEquals("ABC Can | 0.33L", item.getFrontLabel());

@Test  
void testInMemoryDatabaseRetrieval() {  
 Item *item* = inMemoryDatabase.getItemByBarcode("r8yz7clkz4");  
 *assertNotNull*(*item*);  
 *assertEquals*("ABC Can | 0.33L", *item*.getFrontLabel());  
}

1. **TestReceiptProcessing**: Wir fügen einen Artikel hinzu   
    String[] itemData = {"ABC Can | 0.33L", "r8yz7clkz4", "Disposable", "Metal", "0.25"};  
   und versuchen zu sehen, ob wir z. B. in unserem Beleg die gleiche Summe haben wie die Summe der einzelnen Artikel.  
   *assertEquals*("0.25", receiptProcessor.getTotal());

@Test  
void testReceiptProcessing() {  
 String[] *itemData* = {"ABC Can | 0.33L", "r8yz7clkz4", "Disposable", "Metal", "0.25"};  
 receiptProcessor.addItemToReceipt(*itemData*);  
 *assertEquals*("0.25", receiptProcessor.getTotal());

1. **TestDonationFunctionality**: Wir fügen einen Artikel hinzu und tätigen eine Spende. Anschließend prüfen wir, ob die Spendenbasis nicht leer ist bzw dieser Artikel hat assertTrue(donationDatabase.hasDonation());

@Test  
void testDonationFunctionality() {  
 receiptProcessor.addItemToReceipt(new String[]{"ABC Can | 0.33L", "r8yz7clkz4", "Disposable", "Metal", "0.25"});  
 supervisoryModule.donate();  
 // You might want to assert the donation state or database state here  
 *assertTrue*(donationDatabase.hasDonation());  
}

1. TestStateChangeToReady : Wir testen, ob der Pfandautomat den Status richtig ändern kann

@Test  
void testStateChangeToReady() {  
 bottleDepositMachine.changeStateToReady();  
 *assertEquals*(bottleDepositMachine.getCurrentState(), State.*READY*);  
}

## 2. Mockito

Beispiele zur rechtlichen Verbindlichkeitdes von Pfandautomaten:

1. MUSS:
   * *Der Pfandautomat MUSS die LED-Anzeigen für den Betriebsstatus korrekt schalten*.
     + *Erklärung*: Die LED-Anzeigen sind ein wichtiger Bestandteil der Benutzerführung und müssen zuverlässig funktionieren. Anzeige eines falschen Zustands kann eine Kette von Problemen verursachen. Beispielsweise kann die Anzeige von Grün, obwohl Rot angezeigt werden musste, weitere Probleme verursachen und einige davon könnten sogar unerwartet sein.
   * *Der Pfandautomat MUSS den Benutzer benachrichtigen, wenn ein Artikel nicht akzeptiert wird*.
     + *Erklärung*: Die Rückmeldung an den Benutzer ist essenziell, um die Interaktion zu erleichtern und Missverständnisse zu vermeiden.
   * *Der Pfandautomat MUSS jeden eingeworfenen Artikel scannen und prüfen*.
     + *Erklärung*: Ohne die Fähigkeit, Artikel zu scannen und deren Pfandfähigkeit zu prüfen, könnte der PA seine Kernaufgabe nicht erfüllen.
2. SOLLTE**:**
   * *Der Pfandautomat SOLLTE dem Benutzer eine Möglichkeit zur Spende des Pfandbetrags zu Verfügung steht*.
     + *Erklärung*: Die Spendenoption erhöht den Komfort und unterstützt wohltätige Zwecke, ist aber nicht unbedingt erforderlich.
   * *Der Pfandautomat SOLLTE eine Wartungsmeldung an das Personal senden, wenn ein Problem erkannt wird und wenn etwas nicht funktioniert*.
     + *Erklärung*: Diese Funktion ist wünschenswert, aber nicht zwingend. Sie erleichtert jedoch die Wartung und trägt zur Betriebssicherheit bei.
   * *Der Pfandautomat SOLLTE den Benutzer über den verbleibenden Wert seiner Pfandartikel informieren*.
     + *Erklärung*: Diese Information wäre hilfreich, ist jedoch nur eine Komfortfunktion und keine Grundvoraussetzung für die Funktionsweise.
3. WIRD:
   * *Der Pfandautomat WIRD in einem zukünftigen Update das Gewicht des Artikels als weiteren Prüfschritt erfassen können*.
     + *Erklärung*: Diese Funktion wird geplant, um die Genauigkeit der Artikelprüfung zu erhöhen, ist jedoch noch nicht Bestandteil der aktuellen Anforderungen und wenn die Flaschen voll sind, lehnen Sie sie ab.
   * *Der Pfandautomat WIRD in einem zukünftigen Update das Gewicht des Artikels als weiteren Prüfschritt erfassen können*.
     + *Erklärung*: Diese Funktion wird geplant, um die Genauigkeit der Artikelprüfung zu erhöhen, ist jedoch noch nicht Bestandteil der aktuellen Anforderungen.
   * *Der Pfandautomat WIRD zukünftig eine Verbindung zur zentralen Datenbank des Supermarktes herstellen, um Echtzeit-Updates zu erhalten*.
     + *Erklärung*: Diese Funktion dient der besseren Integration in die Infrastruktur des Supermarktes und ist für die Zukunft vorgesehen.

## 3. Funktionale Anforderungen

**Erklärung der funktionalen Anforderungen**: Die funktionalen Anforderungen beschreiben, was das System tun soll, um seinen Zweck zu erfüllen. Sie beschreiben alle Funktionen, die der Einzahlungsautomat ausführen muss, damit er ordnungsgemäß funktioniert und Einzahlungen erfolgreich zurückgegeben und verarbeitet werden können. Die funktionalen Anforderungen sind spezifisch und geben genau an, welche Aktionen der PA durchführen soll.  
  
Beispiele für funktionale Anforderungen:

1. Prüfung und Annahme des Artikels**:**
   * *Der Pfandautomat MUSS in der Lage sein, eingeworfene Artikel zu erkennen und den Barcode zu scannen*.
     + *Erklärung*: Dies ist eine Kernanforderung, da der PA nur Artikel akzeptieren kann, deren Barcode korrekt erfasst wurde.
   * *Der Pfandautomat MUSS prüfen, ob der Artikel vollständig und in einem akzeptablen Zustand ist*.
     + *Erklärung*: Der Zustand des Artikels (unbeschädigt und sauber, und leer) kann die Pfandfähigkeit beeinflussen. Diese Prüfung verhindert Missbrauch und schützt das System vor Schäden.
   * *Der Pfandautomat MUSS den Artikel zurückweisen, wenn der Barcode unbekannt ist*.
     + *Erklärung*: Diese Funktion stellt sicher, dass nur Pfandartikel akzeptiert werden, die in der Datenbank des PA registriert sind.
2. Anzeige und Benutzerinteraktion**:**
   * *Der Pfandautomat MUSS dem Benutzer den aktuellen Status des Automaten über das Display anzeigen*.
     + *Erklärung*: Die Anzeige informiert den Benutzer, ob der PA bereit ist, Artikel anzunehmen, oder ob eine Aktion erforderlich ist.
   * *Der Pfandautomat SOLLTE dem Benutzer die Option bieten, den Pfandbetrag zu spenden oder einen Pfandbon zu erhalten*.
     + *Erklärung*: Diese Option erhöht die Benutzerfreundlichkeit und bietet eine zusätzliche Auswahlmöglichkeit nach Abschluss der Rückgabe.
   * *Der Pfandautomat MUSS eine LED-Anzeige verwenden, um den Betriebsstatus anzuzeigen (grün, gelb oder rot)*.
     + *Erklärung*: Die LED-Anzeige ermöglicht eine schnelle und visuelle Rückmeldung über den aktuellen Zustand des PA.
3. Sortierung und Lagerung von Artikeln**:**
   * *Der Pfandautomat MUSS Einwegdosen komprimieren und in den entsprechenden Sammelbehälter ablegen*.
     + *Erklärung*: Die automatische Komprimierung spart Platz und ermöglicht eine effiziente Lagerung von Einwegdosen im PA.
   * *Der Pfandautomat MUSS Mehrwegflaschen in separaten Behältern sammeln, ohne sie zu komprimieren*.
     + *Erklärung*: Mehrwegflaschen werden nicht komprimiert, da sie erneut verwendet werden sollen und daher unbeschädigt bleiben müssen.
   * *Der Pfandautomat MUSS sicherstellen, dass Plastikflaschen und Glasflaschen getrennt aufbewahrt werden*.
     + *Erklärung*: Eine getrennte Lagerung nach Material erleichtert das Recycling und erhöht die Effizienz.
4. Pfandbetragsberechnung und -ausgabe**:**
   * *Der Pfandautomat MUSS den gesamten Pfandbetrag für die eingeworfenen Artikel berechnen und anzeigen*.
     + *Erklärung*: Die Berechnung des Pfandbetrags ist eine Hauptanforderung, da sie die Grundlage für die Rückgabe an den Benutzer bildet.
   * *Der Pfandautomat SOLLTE dem Benutzer die Möglichkeit bieten, den Pfandbetrag als Spende zu nutzen*.
     + *Erklärung*: Diese Funktion bietet dem Benutzer die Wahl, den Betrag an eine wohltätige Organisation zu spenden.
   * *Der Pfandautomat WIRD in Zukunft die Möglichkeit bieten, den Pfandbetrag kontaktlos auf ein Konto zu überweisen*.
     + *Erklärung*: Diese Funktion ist als zukünftige Erweiterung vorgesehen, um die Nutzerfreundlichkeit zu erhöhen.
5. Benachrichtigungen und Rückmeldungen**:**
   * *Der Pfandautomat MUSS eine Fehlermeldung anzeigen, wenn ein Artikel nicht akzeptiert wird*.
     + *Erklärung*: Eine klare Rückmeldung hilft dem Benutzer, zu verstehen, warum ein Artikel abgelehnt wurde, und verhindert Frustration.
   * *Der Pfandautomat MUSS dem Benutzer die Option bieten, den Vorgang mit einem „Finish“-Button abzuschließen*.
     + *Erklärung*: Der „Finish“-Button signalisiert den Abschluss der Rückgabe und löst die Pfandbetragsberechnung aus.
   * *Der Pfandautomat SOLLTE den aktuellen Wert der akzeptierten Artikel auf dem Display anzeigen*.
     + *Erklärung*: Die Anzeige des Pfandbetrags ermöglicht dem Benutzer eine direkte Kontrolle über den Summewert seiner Rückgabe.

## Konzept des Pfandautomaten (PA)

Das Konzept des Pfandautomaten (PA) wurde unter ökologischen Gesichtspunkten entwickelt. Er soll Pfandflaschen und -dosen korrekt erfassen, umweltschonend verarbeiten und schnell und präzise nach Flaschenart und -größe abrechnen. Dies geschieht im regulären Betrieb des Pfandautomaten, wobei ihm durch unterschiedliche farbige LDE-Flutlichtsignalisierungen der Status Angebot + Anfrage mitgeteilt wird.  
Die wesentlichen Funktionalitäten, die sich daraus ergeben, sind, dass der PA zunächst Pfandteile annehmen kann (grüne LED-Anzeige), dann Art und Größe der Pfandstücke erkennt (mittels Barcodes auf den Pfandflaschen), das Pfand berechnet und einen entsprechenden Beleg (Pfandquittung oder Spendenmöglichkeit zur Übertragung auf ein iPhone oder Smartphone) an den Benutzer oder ggf. an den parkt-Webserver als Input des Komponentensystems zurückgibt.

Hier folgen Beispiele für die verschiedenen Anforderungsschritte (Schritt 0-5), wie sie in der *SOPHIST-Satzschablone* definiert sind:

## Schritt 0: Das System benennen

**Beschreibung**: In diesem Schritt wird das System( Pfandautomat ) klar benannt, damit es eindeutig wird, welches Gerät die jeweilige Funktion ausführen soll.

**Beispiele**:

1. Der *Pfandautomat* MUSS Flaschen und Dosen annehmen und prüfen.
   * *Erklärung*: Hier wird der Pfandautomat (PA) als das System erwähnt. Der PAsoll Flaschen und Dosen verarbeitet.
2. Der *Pfandautomat* SOLLTE eine Spendenoption anbieten.
   * *Erklärung*: Hier soll der Automat die Option bieten, den Pfandbetrag zu spenden, (nicht notwendig, jedoch eine Option).
3. Der *Pfandautomat* MUSS den berechneten Pfandbetrag auf dem Display anzeigen.
   * *Erklärung*: Das Display vermittelt dem Nutzer, wie hoch der Pfandbetrag beträgt, damit der Kunde informiert ist.
4. Der *Pfandautomat* SOLLTE eine Liste akzeptierter Artikel anzeigen.
   * *Erklärung*: Die Anzeige informiert den Nutzer darüber, welche Artikel der Automat aufzunehmen bereit ist.
5. Der *Pfandautomat* MUSS Statusanzeigen in verschiedenen LED-Farben darstellen können.
   * *Erklärung*: Der Automat zeigt seinen Status durch verschiedene LED-Farben an, um Nutzern visuelles Feedback zu geben.
6. @BeforeEach  
   void setUp() {  
    display = new Display();  
    cardScanner = new CardScanner(bottleDepositMachine);  
    bottleDepositMachine = new BottleDepositMachine(display);  
    supervisoryModule = new SupervisoryModule(bottleDepositMachine, cardScanner, display);  
    inMemoryDatabase = new InMemoryDatabase();  
    receiptProcessor = new ReceiptProcessor();  
    donationDatabase = new DonationDatabase();  
   }

## Schritt 1: Die rechtliche Verbindlichkeit festlegenn

**Beschreibung**: Hier wird festgelegt, ob eine Anforderung verpflichtend (MUSS), wünschenswert (SOLLTE), oder zukünftig (WIRD) geplant ist, damit Entwickler und Stakeholder wissen, wie wichtig die Umsetzung der Funktion ist.

**Beispiele**:

1. **MUSS**: Der *Pfandautomat MUSS* jeden Artikel scannen, der eingeworfen wird.
   * *Erklärung*: Anderenfalls wird der Sinn der Hauptaufgabe des Geräts verfehlt.
2. **SOLLTE**: Der *Pfandautomat SOLLTE* die Möglichkeit bieten, den Pfandbetrag zu spenden.
   * *Erklärung*: Die Spendeneinrichtung ist hilfreich und sollte möglichst immer zur Verfügung stehen.
3. **WIRD**: Der *Pfandautomat WIRD* in Zukunft kontaktlose Zahlungen ermöglichen.
   * *Erklärung*: Für die zukünftige Funktion ist es geplant, dass der Pfandautomat kontaktloses Bezahlen unterstützt.
4. **MUSS**: Der *Pfandautomat MUSS* den Kunden bei einem Fehler benachrichtigen.
   * *Erklärung*: Die Fehlermeldung ist eine notwendige Möglichkeit, um den Kunden auf einschlägige Probleme aufmerksam zu machen.
5. **SOLLTE**: Der *Pfandautomat SOLLTE* eine Wartungsmeldung an das Personal senden, falls erforderlich.
   * *Erklärung*: Wartungsmeldung ist zwar keinen dringende Notwendigkeit, jedoch nützlich..

## Schritt 2: Funktionalität identifizieren

**Beschreibung**: In diesem Schritt wird festgelegt, welche Hauptfunktion das System ausführen soll, meist durch Verben wie „annehmen“, „scannen“ oder „berechnen“.

**Beispiele**:

1. Der *Pfandautomat MUSS scannen*, ob ein Artikel akzeptiert wird.
   * *Erklärung*: Ein Scanner ist Teil des Geräts, um die Produkte zu identifizieren.
2. Der *Pfandautomat MUSS dem Kunden die Möglichkeit bieten, eine Spende auszuwählen*.
   * *Erklärung*: Der Automat bietet dem Kunden eine Auswahlmenü.
3. Der *Pfandautomat MUSS anzeigen*, wenn ein Artikel nicht akzeptiert wird.
   * *Erklärung*: Das Display informiert den Kunden, falls ein Artikel nicht akzeptiert wird.
4. Der *Pfandautomat SOLLTE berechnen*, wie viel Pfand der Kunde erhält.
   * *Erklärung*: Hier wird die Berechnung des Pfandbetrags beschrieben.
5. Der *Pfandautomat MUSS prüfen*, ob der Artikel beschädigt ist und nicht angenommen werden kann.
   * *Erklärung*: Der Automat überprüft, ob ein Artikel in akzeptablem Zustand ist.

## Schritt 3: Art der Funktionalität

**Beschreibung**: Hier wird erklärt, wie die Funktion im system abläuft, ob sie automatisch, in Zusammenarbeit mit dem Benutzer oder in Verbindung mit anderen Systemen erfolgt.

**Beispiele**:

1. Der *Pfandautomat MUSS die LED selbstständig auf grün schalten*, sobald er betriebsbereit ist.
   * *Erklärung*: ED zeigt durch Farbwechsel an, dass der Automat einsatzbereit ist.
2. Der *Pfandautomat MUSS dem Kunden die Möglichkeit bieten, den Button „Finish“ zu drücken*.
   * *Erklärung*: Durch Aktivität des Kunden wird der Tauschprozess gestartet und geendet.
3. Der *Pfandautomat SOLLTE Barcodes empfangen und verarbeiten können*.
   * *Erklärung*: Der Automat verarbeitet hier eingehende Daten (Barcodes).
4. Der *Pfandautomat MUSS automatisch prüfen*, ob der Artikel registriert ist.
   * *Erklärung*: Die Registrierung eines Artikels läuft automatich ab.
5. Der *Pfandautomat WIRD dem Kassensystem ermöglichen, den aktuellen Pfandbetrag abzurufen*.
   * *Erklärung*: Diese Funktion ist eine Schnittstelle zum Kassensystem und Verbindung und Übermittlung der Information erfolgt hier.

## Schritt 4: Das Objekt identifizieren

**Beschreibung**: Dieser Schritt legt das zentrale Objekt fest, auf das die Funktion abzielt. Meist sind dies Artikel, Barcodes oder der Pfandbetrag.

**Beispiele**:

1. Der *Pfandautomat MUSS jeden Artikel scannen*.
   * *Erklärung*: „Artikel“ ist hier das zentrale Objekt.
2. Der *Pfandautomat MUSS den Pfandbetrag berechnen*.
   * *Erklärung*: Der berechnete Betrag wird als Resultat geliefert.
3. Der *Pfandautomat SOLLTE dem Kunden den Status anzeigen*.
   * *Erklärung*: Der „Status“ ist die Information, die dem Kunden angezeigt wird.
4. Der *Pfandautomat MUSS die barcode-Informationen prüfen*.
   * *Erklärung*: „Barcode-Informationen“ werden analysiert, um zu erkennen, ob ein Artikel akzeptiert wird.
5. Der *Pfandautomat WIRD den Entsorgungsbehälter verwenden*, um nicht-akzeptierte Artikel zu lagern.
   * *Erklärung*: Der „Entsorgungsbehälter“ ist das zentrale Objekt für nicht-akzeptierte Artikel. Falls der Betreffende einen Artikel fälschlicherweise in das entsprechende Fach eingeführt hat, dann wird er vom Gerät entweder als nicht zulässig erkannt oder wird akzeptiert.

## Schritt 5: Die Bedingungen formulieren

**Beschreibung**: Im letzten Schritt werden Bedingungen für die Ausführung der Funktionen formuliert. Diese können zeitlich oder logisch sein, um den Ablauf klar zu definieren.

**Beispiele**:

1. Sobald der *Pfandautomat* einen Barcode scannt, MUSS er den Status auf „gesperrt“ setzen.
   * *Erklärung*: Der Pfandatuomat hat die Pflicht, den Status auf „gesperrt“ zu setzen, sobald er einen Barcode gelesen hat.
2. Wenn ein Artikel beschädigt ist, MUSS der *Pfandautomat* ihn zurückgeben.
   * *Erklärung*: Ein beschädigter Artikel wird nicht akzeptiert und an den Kunden zurückgegeben und in Statusfarbe auf gelb umgeschaltet werden.
3. Solange der *Pfandautomat* einen Artikel verarbeitet, SOLLTE die LED rot leuchten.
   * *Erklärung*: Zugleich sollte die LED dann, wenn der Pfandautomat mit der Verarbeitung eines Artikels beschäftigt ist, in Statusfarbe auf rot umgeschaltet werden.
4. Sobald der Kunde den „Finish“-Button drückt, MUSS der *Pfandautomat* die Pfandsumme berechnen und anzeigen.
   * *Erklärung*: Der Abschluss des Vorgangs durch den Button ist ein Signal für Anfang von Summe Berechnung und löst die Berechnung aus.
5. Falls ein Artikel nicht erkannt wird, MUSS der *Pfandatuomat* eine Meldung anzeigen und den Status auf „Aktion erforderlich“ setzen.
   * *Erklärung*: Der Kunde wird informiert, wenn ein Artikel nicht erkannt wird.