© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

5.1 Blackbox-Testverfahren

Da bei Blackbox-Verfahren der innere Aufbau des Testobjekts nicht bekannt ist bzw. nicht herangezogen wird, werden Testfälle aus der Spezifikation abgeleitet oder liegen bereits im Idealfall als Teil der Spezifikation vor. Die Verfahren werden auch als spezifikationsbasierte Testverfahren (oder verhaltens- oder verhaltensgesteuerte Testverfahren) bezeichnet, da sie auf der Spezifikation (den Anforderungen bzw. dem Verhalten) basieren. Ein Test mit allen möglichen Eingabewerten und deren Kombination wäre ein vollständiger Test. Dies ist aber wegen der großen Zahl von möglichen Eingabewerten und Kombinationen unrealistisch (s. Abschnitt 2.1.4). Eine sinnvolle Auswahl aus den möglichen Testfällen muss getroffen werden. Dazu gibt es mehrere Verfahren, die im Folgenden vorgestellt werden.

5.1.1 Äquivalenzklassenbildung

Eingabebereiche werden in Äquivalenzklassen unterteilt.

Die Menge der möglichen konkreten Eingabewerte für ein Eingabedatum (ein Parameter des Testobjekts) wird in Äquivalenzklassen – auch als Partitionen bezeichnet – unterteilt. Zu einer Äquivalenzklasse gehören alle Eingabewerte, bei denen der Tester davon ausgeht, dass sich das Testobjekt bei Eingabe eines beliebigen Datums aus der Äquivalenzklasse gleich verhält. Der Test eines Repräsentanten einer Äquivalenzklasse wird als ausreichend angesehen, da davon ausgegangen wird, dass das Testobjekt für alle anderen Eingabewerte derselben Äquivalenzklasse keine andere Reaktion zeigt.

Neben den Äquivalenzklassen, die gültige Eingaben umfassen, sind auch solche für ungültige Eingaben zu berücksichtigen.

Beispiel: Äquivalenzklassenbildung

Das Beispiel zur Berechnung der Rabatte der Autoverkäufer aus Abschnitt 2.3.4 soll hier erneut aufgegriffen werden und den Sachverhalt verdeutlichen. Zur Erinnerung: Über die Verkaufssoftware kann das Autohaus seinen Verkäufern Rabattregeln vorgeben. In der Beschreibung der Anforderungen findet sich folgende Textpassage:

> »Bei einem Kaufpreis von weniger als 15.000 € soll kein Rabatt gewährt werden. Bei einem Preis bis zu 20.000 € sind 5 % Rabatt angemessen. Liegt der Kaufpreis unter 25.000 €, sind 7 % Rabatt möglich, darüber sind 8,5 % Rabatt einzuräumen.«

Es lassen sich sehr einfach vier unterschiedliche Äquivalenzklassen mit gültigen Werten (gültige Äquivalenzklasse, gÄK) für die Berechnung der Rabatte anhand des Verkaufspreises ableiten:

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Tab. 5–1
Gültige Äquivalenzklassen und Repräsentanten

Parameter	Äquivalenzklasse	Repräsentant
Verkaufspreis	$gÄK_1$: $0 \leq x < 15000$	14500
	$gÄK_2$: $15000 \leq x \leq 20000$	16500
	$gÄK_3$: $20000 < x < 25000$	24750
	$gÄK_4$: $x \geq 25000$	31800

In Abschnitt 2.3.4 wurden die Eingabewerte 14500, 16500, 24750 und 31800 (vgl. Tab. 2–3) gewählt. Jeder Wert ist ein Repräsentant aus einer der vier Äquivalenzklassen. Es wird davon ausgegangen, dass Testläufe mit beispielsweise den Eingabewerten 13400, 17000, 22300 und 28900 zu keinen weiteren Erkenntnissen führen, also auch keine weiteren Fehlerwirkungen aufdecken und damit nicht durchgeführt zu werden brauchen. Hinweis: Tests mit Grenzwerten der Äquivalenzklassen (z.B. 15000) werden in Abschnitt 5.1.2 erörtert.

Äquivalenzklassen mit ungültigen Werten

Neben den gültigen Eingabewerten müssen auch ungültige Eingaben beim Test überprüft werden. Es müssen also auch Äquivalenzklassen für ungültige Werte überlegt und Testläufe mit Repräsentanten dieser Klassen durchgeführt werden.

Beispiel

In dem Beispiel sind es die folgenden beiden ungültigen Äquivalenzklassen[3] (uÄK):

Tab. 5–2
Ungültige Äquivalenzklassen und Repräsentanten

Parameter	Äquivalenzklasse	Repräsentant
Verkaufspreis	$uÄK_1$: $x < 0$ (»negativer« – also falscher – Verkaufspreis)	-4000
	$uÄK_2$: $x > 1000000$ (»unrealistisch hoher« Verkaufspreis[a])	1500800

a. Der Wert 1000000 ist hier relativ willkürlich gewählt. Es muss mit dem Automobilhersteller bzw. dem Verkäufer abgestimmt werden, was für ihn ein »unrealistisch hoher« Verkaufspreis ist.

Um die Testfälle systematisch herzuleiten, wird wie folgt vorgegangen: Für jede zu testende Eingabevariable (z.B. Funktions-/Methodenparameter beim Komponententest oder Maskenfeld beim Systemtest) wird der Definitionsbereich ermittelt. Dieser Definitionsbereich ist die Äquivalenzklasse aller zulässigen bzw. erlaubten Eingabewerte. Diese Werte muss das Testobjekt gemäß der Spezifikation verarbeiten. Die Werte

3. Eine korrektere Bezeichnung wäre »Äquivalenzklassen für ungültige Werte« statt ungültige Äquivalenzklassen, da die Äquivalenzklasse ja nicht ungültig ist, sondern die Werte der Äquivalenzklasse in Bezug auf die spezifizierte Eingabe.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

außerhalb des Definitionsbereichs werden als Äquivalenzklassen mit unzulässigen Werten betrachtet. Auch für diese Werte ist zu prüfen, wie das Testobjekt sich verhält.

Weitere Aufteilung der Äquivalenzklassen

Als nächster Schritt sind die Äquivalenzklassen zu verfeinern. Äquivalenzklassenelemente, die vom Testobjekt laut Spezifikation unterschiedlich verarbeitet werden, sind einer neuen (Unter-)Äquivalenzklasse zuzuordnen. Die Äquivalenzklassen werden so lange aufgeteilt, bis sich alle unterschiedlichen Anforderungen mit den jeweiligen Äquivalenzklassen decken. Für jede einzelne Äquivalenzklasse ist dann ein Repräsentant für einen Testfall auszuwählen.

Zur Vervollständigung der Testfälle muss zu jedem Repräsentanten auch das erwartete Ergebnis und ggf. die Vorbedingungen für den Testlauf festgelegt werden.

Äquivalenzklassen für Ausgabe- und weitere Werte

Das Prinzip der Zerlegung kann genauso für die Ausgabewerte genutzt werden. Die Ermittlung der einzelnen Testfälle ist allerdings aufwendiger, da für jeden »Ausgaberepräsentanten« (Ergebniswert) die ihn erzeugenden Eingabewerte zu ermitteln sind. Auch bei den Ausgabewerten sind die Äquivalenzklassen mit ungültigen Werten nicht zu vernachlässigen.

Grundsätzlich können Äquivalenzklassen für jedes zusammenhängende Datenelement in Bezug auf das Testobjekt gebildet werden, also nicht nur für Ein- und Ausgaben. Es können interne Werte, zeitbezogene Werte (z. B. vor oder nach einem Ereignis) und Werte für Schnittstellenparameter, die während Integrationstests getestet werden, zur Bildung von Äquivalenzklassen herangezogen werden.

Es ist darauf zu achten, dass jeder mögliche (Eingabe/Ausgabe-) Wert nur genau zu einer Äquivalenzklasse gehört. Äquivalenzkassen sind überschneidungsfrei, d. h., ein Wert darf nicht zu mehreren Äquivalenzklassen gehören. Die Aufteilung in Äquivalenzklassen und die Wahl der Repräsentanten sind sehr sorgfältig vorzunehmen, da es stark von der Güte der Aufteilung abhängt, welche Testfälle durchgeführt und mit welcher Wahrscheinlichkeit Fehlerwirkungen gefunden werden. In der Regel ist es nicht trivial, die Äquivalenzklassen aus der Spezifikation oder aus anderen Dokumenten herzuleiten.

Grenzen der Äquivalenzklassen

Aussichtsreiche Testfälle sind sicherlich solche, die die Grenzen der Äquivalenzklassen prüfen. Oft sind an diesen Stellen Missverständnisse oder Ungenauigkeiten in den Anforderungen enthalten, da aus einer umgangssprachlichen Formulierung nicht (im mathematischen Sinne) genau hervorgeht, welcher Grenzwert einer Äquivalenzklasse zuzuordnen ist oder nicht. Eine umgangssprachliche Formulierung »... weniger als 15.000 € ...« in den Anforderungen kann den exakten Wert 15.000 innerhalb (ÄK: $x \leq 15000$) oder auch außerhalb der

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Äquivalenzklasse (ÄK: x < 15000) meinen. Ein zusätzlicher Testfall mit x = 15000 kann eine mögliche Fehlinterpretation und damit eine Fehlerwirkung aufdecken. In Abschnitt 5.1.2 wird auf die Analyse der Grenzwerte der Äquivalenzklassen ausführlich eingegangen.

Beispiel: Äquivalenzklassenbildung für ganzzahlige Eingabewerte

Zur Verdeutlichung des Vorgehens bei der Bildung der Äquivalenzklassen sollen die möglichen Äquivalenzklassen für einen ganzzahligen Eingabewert ermittelt werden. Für den ganzzahligen Parameter `extras` der Methode `calculate_price()` ergeben sich folgende Äquivalenzklassen:

Tab. 5–3 *Äquivalenzklassen für ganzzahlige Eingabewerte*

Parameter	Äquivalenzklasse
extras	$gÄK_1$: `[MIN_INT, ... , MAX_INT]` [a] $uÄK_1$: `NaN`

a. Mit MIN_INT und MAX_INT ist jeweils die kleinste und größte ganze darstellbare Zahl im Rechner gemeint. Diese können je nach verwendeter Hardware von Rechner zu Rechner unterschiedlich sein.

Dabei ist darauf zu achten, dass der Wertebereich im Gegensatz zur reinen Mathematik im Rechner durch die Maxima und Minima beschränkt ist. Über- oder Unterschreitungen dieser Werte führen häufig zu Fehlerwirkungen, da mögliche Überschreitungen nicht abgefangen werden.

Die Äquivalenzklasse unzulässiger Werte ergibt sich aus folgender Überlegung: Unzulässige Werte sind Zahlen, die größer oder kleiner sind als die jeweiligen Intervallgrenzen, oder sämtliche nicht numerischen Werte[4]. Wird angenommen, dass die Reaktion der Methode auf einen unzulässigen Wert immer gleich ist (z.B. eine Ausnahmebehandlung, die den Fehlercode NOT_VALID liefert), genügt es, alle möglichen Arten unzulässiger Werte auf eine gemeinsame Äquivalenzklasse (hier bezeichnet mit NaN für »Not a Number«) abzubilden. Zu dieser gemeinsamen Äquivalenzklasse zählen hier auch Gleitkommazahlen, da erwartet wird, dass auf eine Eingabe von beispielsweise »3,5« das Testobjekt mit einer Fehlermeldung reagiert. Die Äquivalenzklassenbildung verlangt in diesem Fall keine weitere Unterteilung, da bei allen falschen Eingaben die gleiche Reaktion erwartet wird. Ein erfahrener Tester wird allerdings auch den Testfall mit einer Gleitkommazahl vorsehen, um festzustellen, ob das Programm nicht möglicherweise eine Rundung der Zahl vornimmt und mit dem dann ermittelten ganzzahligen Wert die Berechnung fortsetzt. Grundlage für diesen ergänzenden Testfall ist demnach das erfahrungsbasierte Testen (s. Abschnitt 5.3).

4. Es hängt von der gewählten Programmiersprache und dem eingesetzten Compiler ab, ob und welche unzulässigen Werte der Compiler erkennt, z.B. beim Versuch, die Methode vom Testtreiber aufzurufen. Im Beispiel wird angenommen, dass der Compiler unzulässige Parameterwerte nicht erkennt und deren Verarbeitung daher im dynamischen Test mit untersucht werden muss.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Da negative und positive Werte erfahrungsgemäß unterschiedlich behandelt werden, macht es Sinn, die Äquivalenzklasse mit den gültigen Werten ($gÄK_1$) weiter aufzuteilen. Auch ist die Null ein Eingabewert, der häufig zu Fehlerwirkungen führt.

Parameter	Äquivalenzklasse	Repräsentant
extras	$gÄK_1$: [MIN_INT,...,0[[a]	-123
	$gÄK_2$: [0,...,MAX_INT]	654
	$uÄK_1$: NaN	"f"

Tab. 5–4 Äquivalenzklassen und Repräsentanten für ganzzahlige Eingabewerte

a. »[« bezeichnet ein offenes Intervall, bei dem die Intervallgrenze nicht dazugehört. Die Intervallangabe [MIN_INT, ..., -1] definiert die gleiche Zahlenmenge, da es sich um ganze Zahlen handelt.

Als Repräsentant wurde relativ willkürlich je ein Wert aus den insgesamt drei Äquivalenzklassen gewählt. Hinzu kommen noch die Grenzwerte (s.a. Abschnitt 5.1.2) der jeweiligen Äquivalenzklassen: MIN_INT, -1, 0, MAX_INT. Für die Äquivalenzklasse der ungültigen Werte gibt es im engeren Sinne keine Grenzwerte.

Damit ergeben sich nach der Äquivalenzklassenmethode mit Berücksichtigung der Grenzwerte für den Test des ganzzahligen Parameters `extras` folgende sieben zu prüfende Werte:

```
{"f", MIN_INT, -123, -1, 0, 654, MAX_INT}.
```

Zu jedem dieser Eingabewerte sind die erwarteten Ausgaben bzw. Reaktionen des Testobjekts festzulegen, um nach dem Testlauf entscheiden zu können, ob ggf. eine Fehlerwirkung vorliegt.

Äquivalenzklassen von Eingabewerten, die keine Basisdatentypen sind

Bei den im Beispiel verwendeten Eingabewerten mit ganzzahligem Wertebereich lassen sich sehr leicht Äquivalenzklassen bilden und die entsprechenden Repräsentanten auswählen. Es können als Wertebereiche außer den elementaren Datentypen auch zusammengesetzte Datenstrukturen oder Objektmengen zugrunde gelegt werden. Es ist dann jeweils zu entscheiden, mit welchen Repräsentanten ein Testfall durchzuführen ist.

Beispiel für Eingabewerte, die aus einer Menge auszuwählen sind

Folgendes Beispiel soll den Sachverhalt verdeutlichen. Ein Kaufinteressent kann ein Erwerbstätiger, ein Student, ein Auszubildender oder ein Rentner sein. Wenn das Testobjekt auf die jeweilige Ausprägung des Kaufinteressenten unterschiedlich reagiert bzw. reagieren soll, dann sind alle Möglichkeiten jeweils mit einem eigenen Testfall zu überprüfen. Wird kein unterschiedliches Verhalten gefordert oder erwartet, kann auch ein Testfall ausreichend sein und ein beliebiger der möglichen Werte für den Kaufinteressenten gewählt werden.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Ist das Testobjekt die Komponente, die die Finanzierung (*EasyFinance*) bearbeitet, so sind vier unterschiedliche Testfälle vorzusehen. Der Finanzrahmen wird bei den verschiedenen Personen (Erwerbstätiger, Student, Auszubildender, Rentner) sicherlich unterschiedlich berechnet. Genaues muss aus den Anforderungen hervorgehen. Jede Berechnung ist einem Test zu unterziehen, um die Korrektheit der Berechnung zu überprüfen und ggf. Fehlerwirkungen nachzuweisen.

Beim Test der Komponente, die die Konfiguration des Fahrzeugs online vornimmt (*DreamCar*), kann es ausreichend sein, dass für einen Käufer nur ein Repräsentant, zum Beispiel ein Erwerbstätiger, gewählt wird. Vermutlich – auch hier muss Genaues in den Anforderungen zum Sachverhalt definiert sein – spielt es bei der Fahrzeugkonfiguration keine Rolle, ob diese von einem Studenten oder von einem Rentner vorgenommen wird. Dem Tester muss aber klar sein, dass er, wenn er nur den Testfall mit dem Eingabewert »Erwerbstätiger« durchführt, keinerlei Aussagen treffen kann, ob die Fahrzeugkonfiguration auch für die anderen Personengruppen ordnungsgemäß durchgeführt wird.

Tipp zur Ermittlung der Äquivalenzklassen

Als Richtlinie für die Ermittlung der Äquivalenzklassen können folgende Hinweise dienen:

- Ermitteln der spezifizierten Einschränkungen und Bedingungen aus der Spezifikation sowohl für die Eingaben als auch für die Ausgaben.
- Für jede Einschränkung bzw. Bedingung ist die Äquivalenzklassenbildung vorzunehmen:
  - Ist ein zusammenhängender Wertebereich spezifiziert, dann sind eine gültige und zwei ungültige Äquivalenzklassen zu berücksichtigen.
  - Ist spezifiziert, dass eine festgelegte Anzahl von Werten (z.B. ein Name aus mindestens 5 und höchstens 7 Zeichen) einzugeben ist, sind eine gültige – mit allen möglichen gültigen Werten – und zwei ungültige Äquivalenzklassen – Unterschreitung (weniger als 5) und Überschreitung (mehr als 7) der gültigen Anzahl – zu bilden.
  - Ist eine Menge von Werten spezifiziert, die möglicherweise unterschiedlich zu behandeln sind, so ist für jeden Wert der Menge eine gültige Äquivalenzklasse (bestehend aus diesem einen Wert) vorzusehen und eine zusätzliche ungültige Äquivalenzklasse.
  - Falls die Einschränkung oder Bedingung eine Situation beschreibt, die zwingend erfüllt werden muss, ist jeweils eine gültige und ungültige Äquivalenzklasse zu berücksichtigen.
- Bestehen Zweifel an der Gleichbehandlung von Werten innerhalb einer Äquivalenzklasse, muss die Äquivalenzklasse entsprechend weiter unterteilt werden.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Testfälle

Regeln zur Testfallerstellung

In der Regel besitzt ein Testobjekt mehr als nur einen Eingabeparameter. Die Äquivalenzklassenmethode liefert für jeden einzelnen dieser Parameter des Testobjekts mindestens zwei Äquivalenzklassen (eine gültige und eine ungültige). Damit gibt es je Parameter mindestens zwei Repräsentanten, die als Testeingaben zu verwenden sind.

Zur Spezifikation eines Testfalls muss jedem Parameter ein Eingabewert zugeordnet werden. Dazu muss entschieden werden, welche der verfügbaren Repräsentanten miteinander zu einem Eingabedatensatz zu kombinieren sind. Um alle (durch die vorgenommene Äquivalenzklassenzerlegung modellierten) Testobjektreaktionen sicher auszulösen, sind die Eingabewerte, also die Repräsentanten der jeweiligen Äquivalenzklassen, nach folgenden Regeln zu kombinieren:

Kombination der Repräsentanten

- Die Repräsentanten aller gültigen Äquivalenzklassen sind zu Testfällen zu kombinieren, d.h., alle möglichen Kombinationen der jeweiligen Repräsentanten sind vorzusehen. Jede dieser Kombinationen bildet einen »gültigen Testfall«.

Ungültige Werte separat testen

- Der Repräsentant einer ungültigen Äquivalenzklasse ist nur mit Repräsentanten von anderen gültigen Äquivalenzklassen zu kombinieren. Für jede ungültige Äquivalenzklasse ist somit ein extra »Negativ«-Testfall zu spezifizieren (s.u.).

Einschränkung der Testfallmenge

Die Anzahl der »gültigen« Testfälle ergibt sich also aus dem Produkt der Zahl gültiger Äquivalenzklassen je Parameter. Wegen dieser multiplikativen Kombination können sich schon bei wenigen Parametern sehr schnell einige Hundert »gültige Testfälle« ergeben. Da es selten machbar ist, so viele Testfälle zu beachten, werden weitere Regeln benötigt, wie die Menge »gültiger« Testfälle reduziert werden kann:

- Die Testfälle aus allen Repräsentanten kombinieren und anschließend nach »Häufigkeit« sortieren (typische Benutzungsprofile). Testfälle in dieser Reihenfolge priorisieren. Zum Test herangezogen werden so nur die »benutzungsrelevanten« Testfälle (oder häufig vorkommende Kombinationen oder Benutzungen).
- Es werden Testfälle bevorzugt, die Grenzwerte oder Grenzwertkombinationen (s. Abschnitt 5.1.2) enthalten.
- Sicherstellen, dass jeder Repräsentant einer Äquivalenzklasse mit jedem Repräsentanten der anderen Äquivalenzklassen in einem Testfall zur Ausführung kommt (d.h. paarweise Kombination statt vollständiger Kombination (s. Abschnitt 5.1.5).

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

- Als Minimalkriterium sicherstellen, dass jeder Repräsentant einer Äquivalenzklasse in mindestens einem Testfall vorkommt.
- Repräsentanten ungültiger Äquivalenzklassen nicht mit Repräsentanten anderer ungültiger Äquivalenzklassen kombinieren.

Ungültige Werte separat testen

Die Repräsentanten ungültiger Äquivalenzklassen werden nicht »multiplikativ« kombiniert. Ein ungültiger Wert soll nur mit »gültigen« kombiniert werden. Denn ein ungültiger Parameterwert löst eine Ausnahmebehandlung aus, unabhängig davon, welche Werte die übrigen Parameter haben. Kombiniert ein Testfall mehrere ungültige Werte, kann es leicht zu einer gegenseitigen Fehlermaskierung kommen, und nur eine der möglichen Ausnahmesituationen wird ausgelöst. Auch ist dann beim Auftreten einer Fehlerwirkung nicht klar, welcher ungültige Wert die Wirkung ausgelöst hat. Vermeidbarer Aufwand zur Fehleranalyse wird dann notwendig.[5]

Beispiel: Test der DreamCar-Preisberechnung

Im Folgenden dient wieder die Methode `calculate_price()` des VSR-II-Teilsystems *DreamCar* als Testobjekt (s. Abschnitt 3.4.1). Zu testen ist, ob die Methode aus ihren Eingabewerten stets einen gemäß der Spezifikation korrekten Gesamtpreis berechnet. Es wird angenommen, dass der innere Aufbau der Methode nicht bekannt ist, sondern nur die funktionale Spezifikation der Methode und die Methodenschnittstelle:

```
double calculate_price (
   double baseprice,       // Grundpreis des Fahrzeugs
   double specialprice,    // Sondermodellaufschlag
   double extraprice,      // Preis der Zusatzausstattung
   int extras,             // Anzahl Zusatzausstattungen
   double discount         // Haendlerrabatt
)
```

Schritt 1
Definitionsbereich ermitteln

Zur Herleitung der nötigen Testfälle aus den Eingabeparametern wird die Äquivalenzklassenbildung eingesetzt. Im ersten Schritt wird für jeden Eingabeparameter sein Definitionsbereich ermittelt. Hieraus resultieren je Parameter Äquivalenzklassen gültiger und ungültiger Werte (s. Tab. 5–5).

5. Manchmal kann es sinnvoll sein, als ergänzende Testfälle auch Repräsentanten von ungültigen Äquivalenzklassen miteinander zu kombinieren, um weitere Fehlerwirkungen zu provozieren.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Parameter	Äquivalenzklasse
baseprice	$gÄK_{11}$: [MIN_DOUBLE, ... , MAX_DOUBLE] $uÄK_{11}$: NaN
specialprice	$gÄK_{21}$: [MIN_DOUBLE, ... , MAX_DOUBLE] $uÄK_{21}$: NaN
extraprice	$gÄK_{31}$: [MIN_DOUBLE, ... , MAX_DOUBLE] $uÄK_{31}$: NaN
extras	$gÄK_{41}$: [MIN_INT, ... , MAX_INT] $uÄK_{41}$: NaN
discount	$gÄK_{51}$: [MIN_DOUBLE, ... , MAX_DOUBLE] $uÄK_{51}$: NaN

Tab. 5–5
Gültige und ungültige Äquivalenzklassen der Parameter der Methode

Damit wurden, alleine aus der Schnittstellenspezifikation, je Parameter eine gültige und eine ungültige Äquivalenzklasse abgeleitet (Testdatengeneratoren gehen entsprechend vor, s. Abschnitt 7.1.2).

Schritt 2
Äquivalenzklassen verfeinern, basierend auf der Spezifikation

Um diese Äquivalenzklassen weiter zu zerlegen, wird Information über die Funktionalität der Methode benötigt. Diese Information liefert die Methodenspezifikation (s. Abschnitt 3.4.1). Aus dieser können folgende testrelevanten Aussagen gefolgert werden:

- Die Parameter 1 bis 3 sind (Fahrzeug-)Preise. Preise sind nicht negativ. Die Spezifikation legt keine Preisgrenzen fest.
- Abhängig vom Wert `extras` berechnet sich der Rabatt auf die Zusatzausstattung (10 %, falls extras ≥ 3, bzw. 15 %, falls extras ≥ 5). `extras` stellt die Anzahl der gewählten Zusatzausstattungsteile dar und ist somit nicht negativ.[6] Die Spezifikation legt keine Anzahlobergrenze fest.
- Der Parameter `discount` ist ein Rabatt, er wird prozentual angegeben und liegt zwischen 0 und 100. Da im Spezifikationstext die Zubehörrabattgrenzen in Prozent angegeben sind, kann der Tester annehmen, dass auch der Händlerrabatt prozentual eingegeben wird. Sicherheit gibt eine Rücksprache beim Kunden.

Lücken in den Anforderungen

Diese getroffenen Überlegungen bzw. Festlegungen basieren nicht nur auf der funktionalen Spezifikation. Vielmehr bringt die Analyse einige Lücken in der Spezifikation ans Licht. Diese Lücken »füllt« der Tester, indem er plausible Annahmen trifft, basierend auf Alltagswissen und seiner Testerfahrung, oder indem er bei Kollegen (Tester oder Entwickler) nachfragt. In Zweifelsfällen ist eine Rücksprache beim Kunden ratsam. Aufgrund der Analyse können die bereits gefundenen Äquivalenzklassen verfeinert werden. Je feiner die Äquivalenzklassenzerlegung, umso genauer wird der Test. Sind alle aus der Spezifikation ersichtlichen Bedingungen für die einzelnen Eingabeparameter und auch das Erfahrungswissen des Testers berücksichtigt, ist die Zerlegung abgeschlossen.

6. Gleitkommazahlen fallen in die Äquivalenzklasse NaN, s. Beispiel: Äquivalenzklassenbildung für ganzzahlige Eingabewerte.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Tab. 5–6
Weitere Aufteilung der Äquivalenzklassen der Parameter der Methode `calculate_price()` mit Repräsentanten

Parameter	Äquivalenzklasse	Repräsentant
baseprice	gÄK$_{11}$: [0,..., MAX_DOUBLE]	20000.00
	uÄK$_{11}$: [MIN_DOUBLE,...,0[uÄK$_{12}$: NaN	-1.00 "abc"
specialprice	gÄK$_{21}$: [0,..., MAX_DOUBLE]	3450.00
	uÄK$_{21}$: [MIN_DOUBLE,...,0[uÄK$_{22}$: NaN	-1.00 "abc"
extraprice	gÄK$_{31}$: [0,..., MAX_DOUBLE]	6000.00
	uÄK$_{31}$: [MIN_DOUBLE,...,0[uÄK$_{32}$: NaN	-1.00 "abc"
extras	gÄK$_{41}$: [0,...,2] gÄK$_{42}$: [3,4] gÄK$_{43}$: [5,..., MAX_INT]	1 3 20
	uÄK$_{41}$: [MIN_INT,...,0[uÄK$_{42}$: NaN	-1 "abc"
discount	gÄK$_{51}$: [0,...,100]	10.00
	uÄK$_{51}$: [MIN_DOUBLE,...,0[uÄK$_{52}$:]100,...,MAX_DOUBLE] uÄK$_{53}$: NaN	-1.00 101.00 "abc"

Als Resultat ergeben sich insgesamt 18 Äquivalenzklassen, 7 für gültige Parameterwerte und 11 für ungültige.

Schritt 3
Repräsentanten auswählen

Um Eingabewerte zu erhalten, muss für jede Äquivalenzklasse ein Repräsentant ausgewählt werden. Dazu kann laut Äquivalenzklassentheorie jeder beliebige Wert einer Äquivalenzklasse herangezogen werden. In der Praxis gelingt die Zerlegung aber selten perfekt. Aus Mangel an Detailinformation, aus Zeitmangel oder einfach weil der Aufwand gescheut wird, wird die Zerlegung auf einer bestimmten Stufe abgebrochen. Einzelne Äquivalenzklassen überschneiden sich eventuell.[7] Bei der Repräsentantenwahl muss daher beachtet werden, dass es innerhalb einer Äquivalenzklasse Werte geben kann, auf die das Testobjekt doch unterschiedlich reagieren könnte, oder Werte, die im Produkteinsatz häufiger vorkommen als andere.

Im Fallbeispiel werden deshalb aus den gültigen Äquivalenzklassen Repräsentanten ausgesucht, die plausible Werte darstellen und im Produkteinsatz vermutlich häufig vorkommende Fälle abdecken. Für die ungültigen Äquivalenzklassen werden möglichst »unkomplizierte« Repräsentanten ausgesucht. Die gewählten Repräsentanten sind in Tabelle 5–6 aufgeführt.

Schritt 4
Testfälle kombinieren

Im nächsten Schritt müssen die Repräsentanten zu Testfällen kombiniert werden. Nach obigen Grundregeln ergeben sich 1×1×1×3×1 = 3 »gültige« Testfälle (durch Kombination der jeweiligen Repräsentanten der gültigen

7. Im Idealfall sind die ermittelten Klassen (wie Äquivalenzklassen in der Mathematik) überschneidungsfrei (disjunkt). Das beschriebene informelle Zerlegungsverfahren garantiert dies aber nicht.

© Dies ist urheberrechtlich geschütztes Material. Bereitgestellt von: HS Bonn-Rhein-Sieg Di, Okt 4th 2022, 11:36

Äquivalenzklassen) und 2+2+2+2+3 = 11 »ungültige« Testfälle (durch separaten Test des jeweiligen Repräsentanten jeder ungültigen Äquivalenzklasse). Insgesamt werden aus den 18 Äquivalenzklassen somit 14 Testfälle abgeleitet (s. Tab. 5–7).

Tab. 5–7 *Testfälle für die Methode* `calculate_price()`

	Parameter					
Test-fall	**baseprice**	**special-price**	**extra-price**	**extras**	**discount**	**result**
1	20000.00	3450.00	6000.00	1	10.00	27450.00
2	20000.00	3450.00	6000.00	3	10.00	26850.00
3	20000.00	3450.00	6000.00	20	10.00	26550.00
4	-1.00	3450.00	6000.00	1	10.00	NOT_VALID
5	"abc"	3450.00	6000.00	1	10.00	NOT_VALID
6	20000.00	-1.00	6000.00	1	10.00	NOT_VALID
7	20000.00	"abc"	6000.00	1	10.00	NOT_VALID
8	20000.00	3450.00	-1.00	1	10.00	NOT_VALID
9	20000.00	3450.00	"abc"	1	10.00	NOT_VALID
10	20000.00	3450.00	6000.00	-1	10.00	NOT_VALID
11	20000.00	3450.00	6000.00	"abc"	10.00	NOT_VALID
12	20000.00	3450.00	6000.00	1	-1.00	NOT_VALID
13	20000.00	3450.00	6000.00	1	101.00	NOT_VALID
14	20000.00	3450.00	6000.00	1	"abc"	NOT_VALID

Bei den gültigen Äquivalenzklassen, die mit einer ungültigen Äquivalenzklasse kombiniert werden, sind bei den Testfällen jeweils die gleichen Repräsentanten verwendet worden, um sicherzustellen, dass nur die Veränderung des einen ungültigen Parameters die Reaktion des Testobjekts bewirkt.

Da vier der fünf Parameter nur jeweils eine gültige Äquivalenzklasse aufweisen, ergeben sich nur wenige »gültige« Testfälle. Es besteht kein Anlass, die Testfallmenge weiter zu reduzieren.

Sind die Testfälle aufgestellt, muss zu jedem Testfall das Sollergebnis ermittelt werden. Im Falle obiger »ungültiger« Testfälle ist dies einfach. Hier muss nur der vom Testobjekt zu generierende Fehlercode eingetragen werden. Bei den »gültigen« Tests muss (z.B. per Tabellenkalkulation) das erwartete Ergebnis berechnet werden.

Tipp

■ In Tabelle 5–7 sind alle Testfälle der Methode `calculate_price()` aufgeführt, zum Beispiel auch der Testfall 10 mit einer negativen Anzahl von Extras. In der Praxis wird ein solcher negativer Wert nicht an die Methode `calculate_price()` weiter gereicht, sondern bereits vorher geprüft und abgefangen (s. Design by Contract [Meyer 13]).