

Testprotokoll

Aufgabe	Testen nach Testplan
Datum Testdurchführung	
Autor / Jahrgang	dhaselberger@tgm.ac.at/N
Tester / Jahrgang	

Äquivalenzklassen

ID	Art NF/GF/FF	Merkmale / Beschreibung
1	NF	Normales Verhalten
2	FF	Dokumentierte mögliche Fehlerfälle und Ausnahmen werden überprüft

Arten: NF ... Normalfall, GF ... Grenzfall/Sonderfall, FF ... Fehlerfall

1. Testfall:

Test-ID / ÄK-ID / CuT	U01	1	TGMLogistics.User.User Konstruktor
Beschreibung	EinE BenutzerIn wird korrekt angelegt		
Testvoraussetzungen	Keine		
Testschritte	Schritt: Konstruktor User mit folgenden Werten: Max Muster, Passwort: XA#*<TGM>b34y Tel.Nr.: +4367712341234, eMail: maxmuster@tgm.ac.at , Adresse: Musterstrasse 1, 1010 Wien, UserPriviledge ADMIN.		
Erwartetes Ergebnis	User korrekt angelegt		

Testdurchführung

Testklasse/-methode	TestClass.java ; void test1()			
Fehlerbeschreibung	Keine Fehler vorhanden			
OK oder Fehlerklasse	minor	major	critical	OK

2. Testfall:

Test-ID / ÄK-ID / CuT	U02	1	TGMLogistics.User.User Konstruktor
Beschreibung	KeinE BenutzerIn wird angelegt, wenn die Tel.Nr. nicht korrekt ist		
Testvoraussetzungen	Keine		
Testschritte	User Max Muster (ADMIN), mit TelNr.: 01234ABC123 und eMail maxmuster@tgm.ac.at		
Erwartetes Ergebnis	NoValidPhoneNumberException wird geworfen		

Testdurchführung

Testklasse/-methode	TestClass.java ; void test2()			
Fehlerbeschreibung	Wenn eine Falsche Telefonnummer eingegeben wird, wird die No ValidPhoneNumberException geworfen			
OK oder Fehlerklasse	minor	major	critical	OK ✓

3. Testfall:

Test-ID / ÄK-ID / CuT	U03	1	TGMLogistics.User.User Konstruktor
Beschreibung	KeinE BenutzerIn wird angelegt, wenn die eMail-Adresse nicht passt		
Testvoraussetzungen	Keine		
Testschritte	User Max Muster (ADMIN), mit eMail max@muster@tgm.ac.at		
Erwartetes Ergebnis	NoValidEmailException wird geworfen		

Testdurchführung

Testklasse/-methode	TestClass.java ; void test3()			
Fehlerbeschreibung	Keine			
OK oder Fehlerklasse	minor	major	critical	✓ OK

4. Testfall:

Test-ID / ÄK-ID / CuT	U04	1	TGMLogistics.User.PaymentMethod Konstruktor
Beschreibung	Eine Zahlungsmethode wird korrekt angelegt.		
Testvoraussetzungen	Keine		
Testschritte	1. Valide Kreditkartennummer, etwa: 5425233430109903 2. Korrupte Kreditkartennummer		
Erwartetes Ergebnis	1. Zahlungsmethode korrekt angelegt 2. NoValidCreditCardException wird geworfen		

Testdurchführung

Testklasse/-methode	TestClass.java ; void test4()			
Fehlerbeschreibung	Keine			
OK oder Fehlerklasse	minor	major	critical	✓ OK

5. Testfall:

Test-ID / ÄK-ID / CuT	U05	1	TGMLogistics.User.User addPaymentMethod	
Beschreibung	Eine Zahlungsmethode kann einem/r UserIn korrekt hinzugefügt werden			
Testvoraussetzungen	User mit ADMIN-Privilege, zweiter User mit CUSTOMER-Privilege erzeugt, Zahlungsmethode erzeugt			
Testschritte	<ol style="list-style-type: none">1. Admin-User legt Zahlungsmethode für Customer an.2. Admin-User legt Zahlungsmethode für Customer mit Kreditkaretnnummer an3. Customer legt selbst eine Zahlungsmethode für sich an.4. Customer legt Zahlungsmethode für Admin an.			
Erwartetes Ergebnis	<ol style="list-style-type: none">1. Zahlungsmethode wird angelegt2. Zahlungsmethode wird angelegt3. Zahlungsmethode wird angelegt4. OperationNotAllowedException wird geworfen			

Testdurchführung

Testklasse/-methode	TestClass.java ; void test5()			
Fehlerbeschreibung	Ein Customer darf nicht die Kreditkarteninformationen eines Admint verändern, trotzdem kann er es.			
OK oder Fehlerklasse	minor	✗ major	critical	OK

6. Testfall:

Test-ID / ÄK-ID / CuT	U06	1	TGMLogistics.OrderLogistics.LogisticsSystem addOrder
Beschreibung	Eine Bestellung kann korrekt hinzugefügt werden		
Testvoraussetzungen	Customer erzeugt, Zwei Shirts mit unterschiedlichen Slogans und Größen erzeugt, Bestellung (Order) des Customers erzeugt, die zwei Shirts werden der Bestellung hinzugefügt		
Testschritte	<ol style="list-style-type: none"> 1. Die Bestellung wird hinzugefügt 2. Dieselbe Bestellung wird erneut hinzugefügt 		
Erwartetes Ergebnis	<ol style="list-style-type: none"> 1. Bestellung korrekt hinzugefügt, Status der Bestellung ist PROCESSING 2. Bestellung wird nicht hinzugefügt 		

Testdurchführung

Testklasse/-methode	TestClass.java ; void test6()			
Fehlerbeschreibung	Keine Fehler			
OK oder Fehlerklasse	minor	major	critical	✓ OK

7. Testfall:

Test-ID / ÄK-ID / CuT	U07	1	TGMLogistics.OrderLogistics.LogisticsSystem processOrders
Beschreibung	Eine Bestellung wird nur verschickt, wenn genug vorhanden ist		
Testvoraussetzungen	<ul style="list-style-type: none"> • Customer erzeugt, Store erzeugt, • Zwei Shirts mit unterschiedlichen Slogans und Größen erzeugt, • Shirts werden dem Store zugewiesen (addShirt), • Zwei Bestellungen (Order) des Customers werden erzeugt - mit jeweils zwei Shirts: <ol style="list-style-type: none"> 1. Bestellung: dieselben Slogans und Größen, wie die Shirts im Store; 2. Bestellung: nur ein Shirt gleicht einem im Store, <ul style="list-style-type: none"> • Store und Customer werden dem Logistik-Zentrum zugewiesen 		
Testschritte	Verarbeite die Bestellungen		
Erwartetes Ergebnis	<ol style="list-style-type: none"> 1. OrderStatus der ersten Bestellung ist OrderStatus.DISPATCHED 2. OrderStatus der zweiten Bestellung ist OrderStatus.CANCELLED 		

Testdurchführung

Testklasse/-methode	TestClass.java ; void test7()			
Fehlerbeschreibung	Der Status der 1. Bestellung stimmt nicht mit den Vorgaben überein => Expected DISPATCHED got PROCESSING			
OK oder Fehlerklasse	minor	major	X critical	OK

8. Testfall:

Test-ID / ÄK-ID / CuT	U08	1	TGMLogistics.OrderLogistics.LogisticsSystem removeOrder
Beschreibung	Bestellungen sind nur von autorisierten NutzerInnen löscher		
Testvoraussetzungen	User mit ADMIN-Privilege, zweiter User mit COWORKER-Privilege erzeugt, Zwei Shirts mit unterschiedlichen Slogans und Größen erzeugt, Bestellung (Order) des Customers erzeugt , Bestellung der Bestellliste hinzugefügt (addOrder)		
Testschritte	<ol style="list-style-type: none"> 1. Der ADMIN-User wird als Authorizer übergeben 2. Der ADMIN-User wird als Authorizer mit falschem Passwort übergeben 3. Der COWORKER-User wird als Authorizer übergeben 		
Erwartetes Ergebnis	<ol style="list-style-type: none"> 1. Order wird gelöscht 2. OperationNotAllowedException wird geworfen 3. OperationNotAllowedException wird geworfen 		

Testdurchführung

Testklasse/-methode	Klasse TestClass.java ; void test8()			
Fehlerbeschreibung	Keine Fehler vorhanden			
OK oder Fehlerklasse	minor	major	critical	OK ✓