

SWE1 - Aufgabe 8

Testing

1 Aufgabe 1

Erstellen Sie einen möglichst vollständigen Blackbox-Test für das Interface IChatServer. Die genaue Spezifikation ist in den Kommentaren im Interface gegeben. Das Interface und den ChatServer, der das Interface implementiert, finden Sie im beiliegenden Source-Code.

1.1 Black Box Test: IChatServer

Methode	Äquivalenzklasse	Beispiel
publiziere Nachricht	Gültiger Wert: Eine vom Benutzer eingegebene Nachricht	"Hello, World!"
	Ungültiger Wert: Eine leere Nachricht ohne Inhalt	"" / null
loginChatClient	Gültiger Wert: Ein Benutzername, der nicht leer ist und noch nicht verwendet wird	fluelsim
	Ungültige Werte: - Leerer Benutzername - Benutzername, der bereits verwendet wird	"" / null hans / hans (zweimal der gleiche Namen)
logoutChatClient	/	/
getAnzahlChatRaeume	/	/
raumExistiert	Gültiger Wert: Id eines Raumes der existiert	1
	Ungültiger Wert: Id eines Raumes, welcher nicht existiert	99999
wechsleChatRaum	Gültiger Wert: Id eines Raumes, welches existiert und nicht der Eingangsraum ist	Id = 2
	Ungültiger Wert: - Id eines Raumes, welcher nicht existiert - Id des Hauptraumes	Id = 0 / 99999
getMaxAnzChatters	/	/

1.2 Testprotokoll

ID	Testling	Ablauf	Eingabe	Erwartetes Resultat
1	publiziereNachricht	Es wird eine Nachricht eingegeben, welche anschliessend im Chatraum des Clients angezeigt wird	Es wird die Nachricht "Hello World" gesendet	Im Chatraum in dem sich der Benutzer befindet wird die Nachricht angezeigt
			Es wird eine leere Nachricht gesendet	Im Chatraum wird keine neue Nachricht angezeigt
2	loginChatClient	Es wird ein Nickname gewählt und sich auf dem Server eingeloggt	Es wird als Benutzername "fluelsim" eingegeben und sich damit eingeloggt	Wenn der Nickname noch nicht verwendet wurde, ist der Benutzer auf dem Server eingeloggt
			Es wird kein Benutzername eingegeben	Es erscheint eine Fehlermeldung
3	logoutChatClient	Es wird der Button fürs Logout geklickt	-	Der Client wird vom Server abgemeldet
4	getAnzahlChatRaeume	Es wird die Anzahl der aktuell verfügbaren Chaträume abgefragt	-	Die Anzahl der verfügbaren Chaträume wird angezeigt
5	raumExistiert	Prüfen, ob ein Raum existiert, indem ein Raum übergeben wird	Id eines Raumes der existiert	true
			Id eines Raumes der nicht existiert	false
6	wechsleChatRaum	Der Benutzer wechselt in einen anderen Chat-Raum	Id ($\neq 0$) eines Raumes	<ul style="list-style-type: none"> - Nachricht im alten Raum (Benutzer xy hat den Raum verlassen) - Chatter-Liste im alten Raum ist aktualisiert - Benutzer erhält Willkommensmeldung im neuen Raum - Chatter-Liste im neuen Raum ist aktua-

		Raum-Id (= 0)	lisiert
7	getMaxAnzChatters	Benutzer fragt die maximale Anzahl Chatter ab, die angemeldet sein können	Der Benutzer sieht eine Fehlermeldung, dass er den Eingangsraum nicht mehr betreten kann
		-	Benutzer sieht die maximale Anzahl Chatter, die gleichzeitig angemeldet sein können

2 Aufgabe 2

Schreiben Sie für die Methoden `loginChatClient`, `publiziereNachricht` und `wechsleToChatRaum` des `IChatServer` je einen Satz von JUnit-Tests, der die drei Methoden anhand des Blackbox-Tests aus Aufgabe1 möglichst vollständig testet. Erweitern Sie die gegebene Klasse `ChatServer` so, dass sie die JUnit-Tests für diese drei Methoden besteht. Setzen Sie für die `ChatClients` Mock-Objekte ein, die das Interface `IChatClient` erfüllen. Verwenden Sie dafür das `EasyMock`-Framework.

Siehe Java-Code.