

# Test funzionale (prova orale del 23/01/2020)

## [Individuazione delle categorie]

La funzione **generaReport()** di cui si vogliono determinare i possibili casi di test, accetta come unico parametro un intero (che chiameremo “N”, come da traccia) rappresentante il numero minimo di spese effettuate dal cliente affinché egli possa comparire nel report.

L’output di questa funzionalità è composto di una lista di coppie (Cliente, InfoSpese), ove “InfoSpese” è una “classe di supporto” i cui membri privati sono, rispettivamente, la quantità di spese effettuate dal singolo cliente e l’importo totale speso dal cliente (si considerano, naturalmente, solo le spese in stato “CONSEGNATA”).

Due possibili categorie per il testing sono, quindi:

- Numero minimo di spese;
- Numero di spese consegnate (in rapporto al numero di spese totali effettuate per un determinato cliente, ad esempio Erich, nel metodo **setup()**)

Si riportano, di seguito, delle possibili classi di equivalenza per ognuna delle due categorie:

NumeroMinimoSpese	SpeseConsegnate
<ul style="list-style-type: none"><li>• <math>N &lt; 0</math> [ERR]</li><li>• <math>N = 0</math></li><li>• <math>0 &lt; N &lt; \text{numSpeseClienteMaxSpese}</math></li><li>• <math>N = \text{numSpeseClienteMaxSpese}</math></li><li>• <math>N &gt; \text{numSpeseClienteMaxSpese}</math></li></ul>	<ul style="list-style-type: none"><li>• <math>= \text{speseTotaliCliente}</math></li><li>• <math>&lt; \text{speseTotaliCliente}</math> [SINGLE]</li></ul>

## [Casi di test]

Test case ID	Descrizione	Precondizioni	Input	Output Attesi	Post. Att.	Output Ottenuti	Post. Ott.	Esito
1	Si cercano i clienti che hanno effettuato dalle -1 spese in su	--	generaReport (-1);	Messaggio di errore: “Nessun cliente può aver effettuato un numero negativo di spese”	N/A	Messaggio di errore: “Nessun cliente può aver effettuato un numero negativo di spese.”	N/A	PASS
2	Si cercano i clienti che hanno effettuato dalle 0 spese in su	<b>Erich:</b> 3 spese effettuate, 2 consegnate.  <b>Richard:</b> 2 spese	1 – Consegna una spesa ad Erich	Erich, 3, 70 Richard, 2, 84 Ralph, 0, 0 John, 1, 5	N/A	Erich, 3, 70 Richard, 2, 84 Ralph, 0, 0 John, 1, 5	N/A	PASS

		effettuate, 2 consegnate.  <b>Ralph:</b> 0 spese effettuate, 0 consegnate.  <b>John:</b> 1 spesa effettuata, 1 consegnata.	2 – generaReport (0);					
3	Si cercano i clienti che hanno effettuato dalle 2 spese in su	<b>Erich:</b> 3 spese effettuate, 2 consegnate.  <b>Richard:</b> 2 spese effettuate, 2 consegnate.  <b>Ralph:</b> 0 spese effettuate, 0 consegnate.  <b>John:</b> 1 spesa effettuata, 1 consegnata.	1 – Consegna una spesa ad Erich  2 – generaReport (2);	Erich, 3, 70 Richard, 2, 84	N/A	Erich, 3, 70 Richard, 2, 84	N/A	<b>PASS</b>
4	Si cercano i clienti che hanno effettuato dalle 3 spese in su (Erich ha tutte le spese consegnate)	<b>Erich:</b> 3 spese effettuate, 2 consegnate.  <b>Richard:</b> 2 spese effettuate, 2 consegnate.  <b>Ralph:</b> 0 spese effettuate, 0 consegnate.  <b>John:</b> 1 spesa effettuata, 1 consegnata.	1 – Consegna una spesa ad Erich  2 – generaReport (3);	Erich, 3, 70	N/A	Erich, 3, 70	N/A	<b>PASS</b>
5	Si cercano i clienti che hanno effettuato dalle 3 spese in su (Erich ha solo due spese consegnate)	<b>Erich:</b> 3 spese effettuate, 2 consegnate.  <b>Richard:</b> 2 spese effettuate, 2 consegnate.  <b>Ralph:</b> 0 spese effettuate, 0 consegnate.	generaReport (3);	Nessun cliente trovato (report vuoto)	N/A	Nessun cliente trovato (report vuoto)	N/A	<b>PASS</b>

		<b>John:</b> 1 spesa effettuata, 1 consegnata.						
--	--	--	--	--	--	--	--	--