

<i>Methode</i>	<i>#Tests</i>	<i>#Fehler</i>	<i>Voll. Abd.</i>
testParticipantGeneration()	1	0	Ja
testGeneratePartyLocationFromCSV()	1	0	Ja
testGeneratePartyLocationFromCSV()	4	0	Ja
testGenerateParticipantsAndPairsFromCSV()	1	0	Ja
testGenerateParticipantsAndPairsFromInvalidCSV()	1	0	Ja
testGenerateParticipantsAndPairsFromEmptyCSV()	1	0	Ja
testIsCoupleRegistration()	2	0	Ja

1. Generating Participants and Pairs from CSV:

Vorbedingung: die CSV-Datei *testTeilnehmerliste.csv* existiert und enthält korrekte Daten für mehrere Teilnehmer, einschließlich Paarungen.

Ablauf: zuerst wird die Methode *generateParticipantsAndPairsFromCSV* mit dem Pfad zur CSV-Datei sowie leeren Instanzen von *ParticipantList* und *PairList* aufgerufen. Dann die Methode liest die Daten und füllt die Listen entsprechend den Daten in der CSV-Datei.

Erwartetes Verhalten: die Teilnehmerliste und die Paarliste sind korrekt gefüllt basierend auf den Daten der CSV-Datei

Tatsächliches Verhalten:

2. Generate Participant:

Vorbedingung: die CSV-Datenzeile ist korrekt formatiert und beinhaltet alle notwendigen Informationen für einen Teilnehmer.

Ablauf: Die Methode *createParticipant* wird mit einem String-Array aufgerufen, das die Teilnehmerdaten enthält.

Erwartetes Verhalten: ein Teilnehmerobjekt wird erfolgreich erstellt mit den entsprechenden Attributen, die aus der CSV-Datenzeile gelesen wurden.

Tatsächliches Verhalten:

3. Couple registration :

Vorbedingung String-Arrays mit Daten, die entweder eine Paaranmeldung darstellen oder nicht.

Ablauf: 1. Die Methode *isCoupleRegistration* wird mit String-Arrays aufgerufen, die potenziell Daten für Paare enthalten.

Erwartetes Verhalten Die Methode gibt *true* zurück, wenn es sich um eine Paaranmeldung handelt, andernfalls *false*.

Tatsächliches Verhalten