Methode	#Tests	#Fehler	Voll Abd.
printGroupListQualityTest()	1	0	Ja
removeGroupsOverMaxAllowedLimitTest()	1	0	Ja
finalizeGroupListTest()	1	0	Ja
setupRandomGeneratedGroupListTest()	2	0	Ja
calculateFitnessTest()	1	0	Ja
printEachClusterFitnessTest()	1	0	Ja

1. Print GroupList Quality:

Vorbedingung: allGroups und successorPairList sind initialisiert.
Ablauf: Druckt die Anzahl der Gruppen, Nachrückerpaare, die durchschnittliche Gender-Diversität, Altersdifferenz, Vorliebenabweichung, durchschnittliche Weglänge und Standardabweichung der Weglängen auf der Konsole.
Erwartetes Verhalten: Gibt die genannten Qualitätsmetriken korrekt berechnet auf der Konsole aus.
Tatsächliches Verhalten:

2. Remove Groups over max allowed Limit:

Vorbedingung: maxAllowedGroups ist eine positive ganze Zahl. groupClusterList und allGroups enthalten mindestens eine Gruppe.
Ablauf: Entfernt wiederholt das schlechteste Cluster aus groupClusterList, fügt die Paare des entfernten Clusters zur successorPairList hinzu und entfernt deren Gruppen aus allGroups, bis die Anzahl der Gruppen unter der maximal erlaubten Grenze liegt.
Erwartetes Verhalten: Reduziert die Anzahl der Gruppen in allGroups und groupClusterList auf maximal maxAllowedGroups. Fügt Paare aus entfernten Gruppen zur successorPairList hinzu.
Tatsächliches Verhalten:

3. Finalize GroupList:

Vorbedingung: groupClusterList enthält mindestens einen GroupCluster. Jede Gruppe im Cluster hat genau einen Gastgeber.
Ablauf: Weist jeder Gruppe eine eindeutige ID zu und fügt sie zu allGroups hinzu; setzt für jedes Paar in der Gruppe die entsprechende Gruppen-ID und das hostingDish-Attribut basierend auf der Gruppenzuordnung.
Erwartetes Verhalten: Jede Gruppe in allGroups erhält eine eindeutige ID. Jedes Paar in jeder Gruppe erhält die entsprechende Gruppen-ID und das hostingDish-Attribut.
Tatsächliches Verhalten:

4. Set up random generated GroupList:

Vorbedingung: allPairs ist eine nicht-leere Liste von Paaren. partyLocation und criteriaRankingList sind gesetzt.
Ablauf: Erstellt eine zufällige Gruppierung der Paare in allPairs, basierend auf der Anzahl benötigter Cluster, fügt die restlichen Paare zur successorPairList hinzu, sortiert die Cluster nach Fitness und speichert sie in groupClusterList.
Erwartetes Verhalten: Erstellt GroupCluster mit jeweils 9 Paaren. Die restlichen Paare werden zur successorPairList hinzugefügt. groupClusterList wird nach Fitness sortiert und gesetzt.
Tatsächliches Verhalten:

5. Calculate Fitness:

Vorbedingung: groupClusterList enthält mindestens einen GroupCluster.
Ablauf: Sortiert groupClusterList nach der Fitness der Cluster, summiert die Fitness-Werte der Cluster und setzt den Gesamt-Fitness-Wert für die GroupList.
Erwartetes Verhalten: Die Gesamt-Fitness der GroupList wird korrekt berechnet und gesetzt. groupClusterList bleibt sortiert nach der Fitness der Cluster.
Tatsächliches Verhalten:

6. Print each Cluster Fitness:

Vorbedingung: groupClusterList enthält mindestens einen GroupCluster.
Ablauf: Iteriert durch die groupClusterList und druckt den Fitness-Wert jedes Clusters.
Erwartetes Verhalten: Der Fitness-Wert jedes Clusters wird auf der Konsole ausgegeben.
Tatsächliches Verhalten: