**Algorithmus zur Pärchen- und Gruppen Erstellung**

public void makeBestPairList() {

while (dataList.unmatchedParticipants.size() > 1) {­­­

// Create pairs based on criteria

BestPair = makeBestPair();

// put BestPair to an tempPairList and sort() this list

// Add highest Score pairs to pairList

// Remove paired participants from unmatched list

// Remove Pair that contains Participant from BestPair

}

// Handle remaining unmatched participants

}

public Pair makeBestPair(Participant participant) {

// Find the best pair for a participant based on criteria

// chose the second Participant to make a Pair

// bestpair has the highest match score

}

public void makeBestGroupList() {

while (pairListTemp.size() > 2) {

// Create groups based on pairs

BestGroup = makeBestGroup(Pair)

// Remove paired pairs from temporary pair list

// Add groups to group list

}

If(courseCounter==3){ // Handle Exception in Gang 3

makeBestRestGroupGang3(allCookedGroup,GroupdidntCookGroup);

}

// Handle remaining pairs

}

public Pair makeBestGroup(Pair pair) {

// Find the best Group for a pair based on criteria

// chose the second and third Pair to make a Group

// bestGroup has the highest match score

}

public List<Group> makeBestRestGroupGang3(Group allCookedGroup, Group didntCookGroup) {

// Create the best group for the remaining pairs in Gang 3

//exchange the Pair between

//Group1: every Pair, that have cooked in Gang 1 or 2 and

//Group2: every Pair, that havent cooked after Gang 3

//-> create the rest Group for Gang 3

}