**Algorithmus zur Pärchen- und Gruppen Erstellung**

public void makeBestPairList() {

//divide all the Participants with(-out) Kitchen into 2 List

While(noKitchenParticipants.size() > 0){

//find best Partner with Kitchen and save to the List of Pair

}

While(withKitchenParticipants.size() > 1){

//find best Partner without(or also with) Kitchen and save to the List of Pair

}

//handle the remaining Participants, save to the successor list

}

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

}

while (unmatchedPairCourse3.size() >= 2 && courseCounter==3) {

//find all the Pair which have already cooked after 2 Courses and Pair that didn’t cook at all, //save them to 2 different list and try to build the group from them

}

// Handle remaining pairs

}

public Group makeBestGroup(Pair pair) {

// check all the group criteria and sort out the perfect group

// 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 Group makeBestGroupCourse3((Pair pair, List<Pair> unmatchedPair){

//find the best group for Pair by chosing 2 another Pair in unmatchedPair List

}