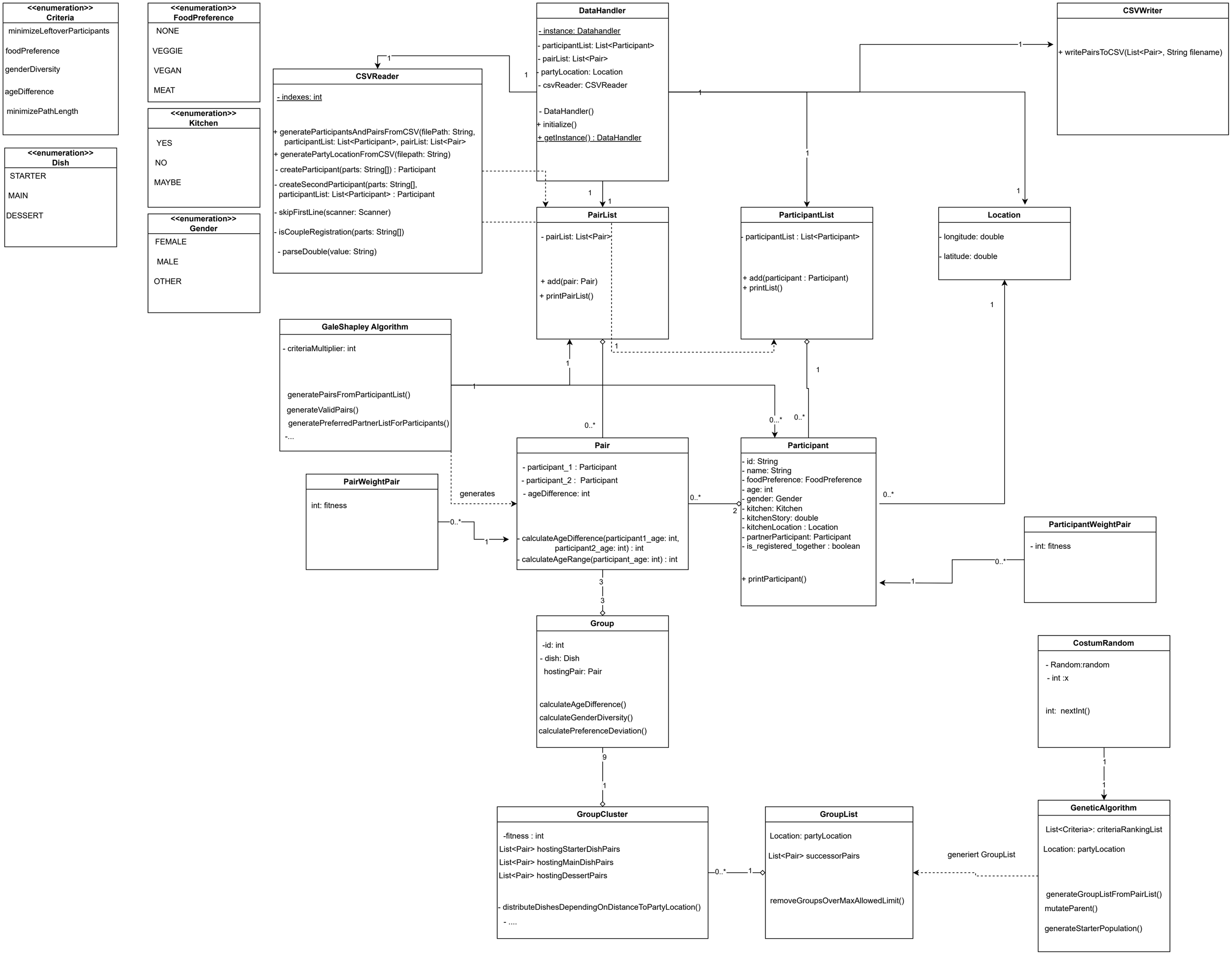
<<enumeration>>
Criteria
minimizeLeftoverParticipants
foodPreference
genderDiversity
ageDifference
minimizePathLength
<<enumeration>>
FoodPreference
NONE
VEGGIE
VEGAN
MEAT
<<enumeration>>
Kitchen
YES
NO
MAYBE
<<enumeration>>
Gender
FEMALE
MALE
OTHER
<<enumeration>>
Dish
STARTER
MAIN
DESSERT
DataHandler
- instance: Datahandler
- participantList: List<Participant>
- pairList: List<Pair>
- partyLocation: Location
- csvReader: CSVReader
- DataHandler()
+ initialize()
+ getInstance() : DataHandler
CSVWriter
+ writePairsToCSV(List<Pair>, String filename)
CSVReader
- indexes: int
+ generateParticipantsAndPairsFromCSV(filePath: String, participantList: List<Participant>, pairList: List<Pair>
+ generatePartyLocationFromCSV(filepath: String)
- createParticipant(parts: String[]) : Participant
- createSecondParticipant(parts: String[], participantList: List<Participant> : Participant
- skipFirstLine(scanner: Scanner)
- isCoupleRegistration(parts: String[])
- parseDouble(value: String)
PairList
- pairList: List<Pair>
+ add(pair: Pair)
+ printPairList()
ParticipantList
- participantList : List<Participant>
+ add(participant : Participant)
+ printList()
Location
- longitude: double
- latitude: double
GaleShapley Algorithm
- criteriaMultiplier: int
generatePairsFromParticipantList()
generateValidPairs()
generatePreferredPartnerListForParticipants()
-...
PairWeightPair
int: fitness
generates
Pair
- participant_1 : Participant
- participant_2 : Participant
- ageDifference: int
- calculateAgeDifference(participant1_age: int, participant2_age: int) : int
- calculateAgeRange(participant_age: int) : int
Participant
- id: String
- name: String
- foodPreference: FoodPreference
- age: int
- gender: Gender
- kitchen: Kitchen
- kitchenStory: double
- kitchenLocation : Location
- partnerParticipant: Participant
- is_registered_together : boolean
+ printParticipant()
ParticipantWeightPair
- int: fitness
Group
-id: int
- dish: Dish
hostingPair: Pair
calculateAgeDifference()
calculateGenderDiversity()
calculatePreferenceDeviation()
CostumRandom
- Random:random
- int :x
int: nextInt()
GroupCluster
-fitness : int
List<Pair> hostingStarterDishPairs
List<Pair> hostingMainDishPairs
List<Pair> hostingDessertPairs
- distributeDishesDependingOnDistanceToPartyLocation()
- ....
GroupList
Location: partyLocation
List<Pair> successorPairs
removeGroupsOverMaxAllowedLimit()
generiert GroupList
GeneticAlgorithm
List<Criteria>: criteriaRankingList
Location: partyLocation
generateGroupListFromPairList()
mutateParent()
generateStarterPopulation()