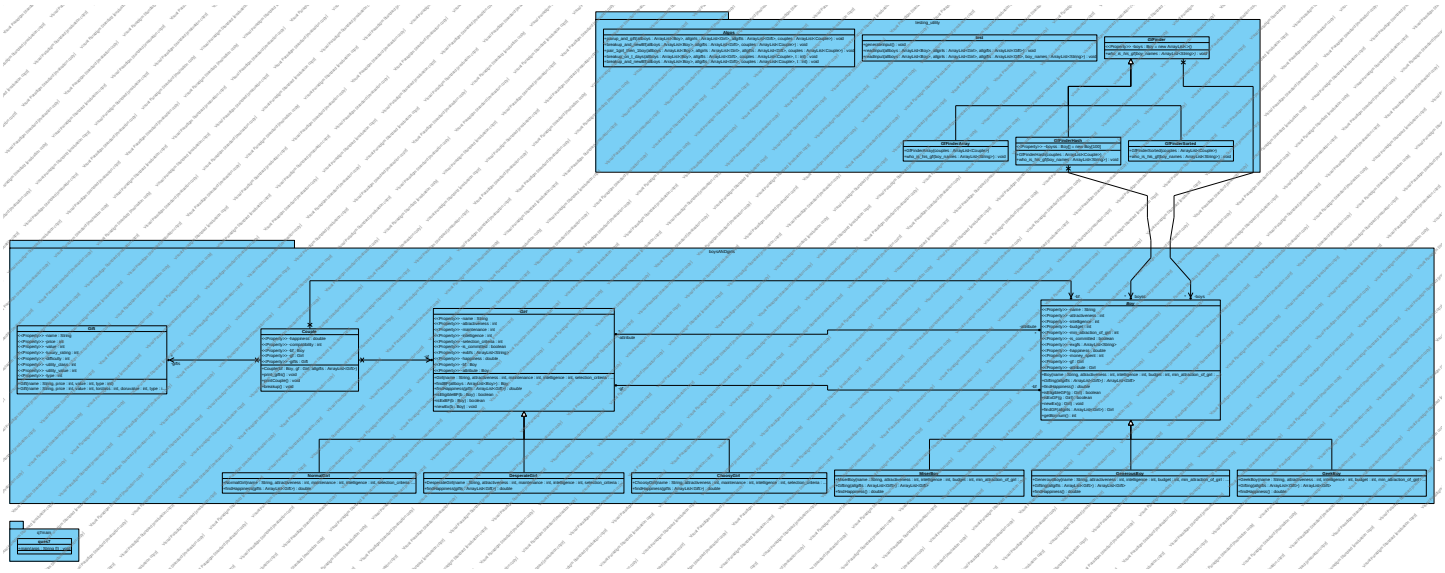
Main
+pickup_and_sell(Bag[] a, Array<int> alights, Array<int> dist, Array<int> boards, Array<Coupler> c) : void
+breakup_and_sell(Bag[] a, Array<int> alights, Array<int> dist, Array<Coupler> c) : void
+sell_bp(Bag[] a, Array<int> alights, Array<int> dist, Array<int> alight, Array<int> board, Array<int> boards, Array<Coupler> c) : void
+breakup_and_sell(Bag[] a, Array<int> alights, Array<int> dist, Array<int> alight, Array<int> board, Array<int> boards, Array<Coupler> c, Bag : int) : void
Bid
+generateBids() : void
+sortBidsByBids(Bag[] a, Array<int> alights, Array<int> dist) : void
+bidBypickBids(Bag[] a, Array<int> alights, Array<int> dist, Array<int> bag_names, Array<int> boards) : void
DPCoupler
+<<Property>> <> boys : Bag[] = new Array<int>()
+sort_a_no_gifting_names : Array<int> boards) : void
DPPrunerAstar
+DPPrunerAstar(pupler : Array<int> c, Array<int> boards) : void
+sort_a_no_gifting_names : Array<int> boards) : void
DPPrunerDijkstra
+<<Property>> <> boys : Bag[] = new Bag[100]
+DPPrunerDijkstra(pupler : Array<int> c, Array<int> boards) : void
+sort_a_no_gifting_names : Array<int> boards) : void
DPPrunerBanked
+DPPrunerBanked(pupler : Array<int> c, Array<int> boards) : void
+sort_a_no_gifting_names : Array<int> boards) : void
boysAndGifts
Gift
+<<Property>> <> name : String
+<<Property>> <> gave : gift
+<<Property>> <> value : int
+<<Property>> <> sysu_rating : int
+<<Property>> <> gifting_prob : int
+<<Property>> <> utility_gave : int
+<<Property>> <> utility_value : int
+<<Property>> <> type : int
+Gift(name : String, price : int, value : int, type : int)
+Gift(name : String, price : int, value : int, budget : int, demand_gif : int, type : i...
Coupler
+<<Property>> <> happiness : double
+<<Property>> <> probability : int
+<<Property>> <> Bag
+<<Property>> <> Gift
+<<Property>> <> gifts : Gift
+Coupler(Bag : Bag, gf : Gift, lefgifts : Array<int>, dist : int)
+sort_gifts() : void
+splitCoupler() : void
+breakup() : void
Bid
+<<Property>> <> name : String
+<<Property>> <> attractiveness : int
+<<Property>> <> maintenance : int
+<<Property>> <> intelligence : int
+<<Property>> <> selection_criteria : int
+<<Property>> <> is_controlled : boolean
+<<Property>> <> width : Array<int>[String]
+<<Property>> <> happiness : double
+<<Property>> <> ID : Bag
+<<Property>> <> bidded : Bag
+Bid(name : String, attractiveness : int, maintenance : int, intelligence : int, selection_criteria : int)
+bidBetween(a : Array<int>) : Bag
+bidHappiness(gifts : Array<int>[int]) : double
+isBidgiftedBy(Bag : Bag) : boolean
+setBid(Bag : Bag) : void
1..*
<<Bid>>
1..*
<<Bid>>
Bag
+<<Property>> <> place : String
+<<Property>> <> attractiveness : int
+<<Property>> <> intelligence : int
+<<Property>> <> budget : int
+<<Property>> <> min_attraction_of_gif : int
+<<Property>> <> is_controlled : boolean
+<<Property>> <> bidded_gifts : Array<int>
+<<Property>> <> happiness : double
+<<Property>> <> money_spent : int
+<<Property>> <> gf : int
+<<Property>> <> attribute : Dict
+Bag(name : String, attractiveness : int, intelligence : int, budget : int, min_attraction_of_gif : int)
+setBaggifts(gifts : Array<int>[int]) : Array<int>[int]
+isBidgifted(Bid : Bid) : boolean
+isBidFrg(Bid : Bid) : boolean
+setBid(Bid : Bid) : void
+BidGiftGiftgifts : Array<int>[int] : Dict
+getBid(Bid) : int
1..*
<<Bid>>
1..*
<<Bid>>
1..*
<<Bid>>