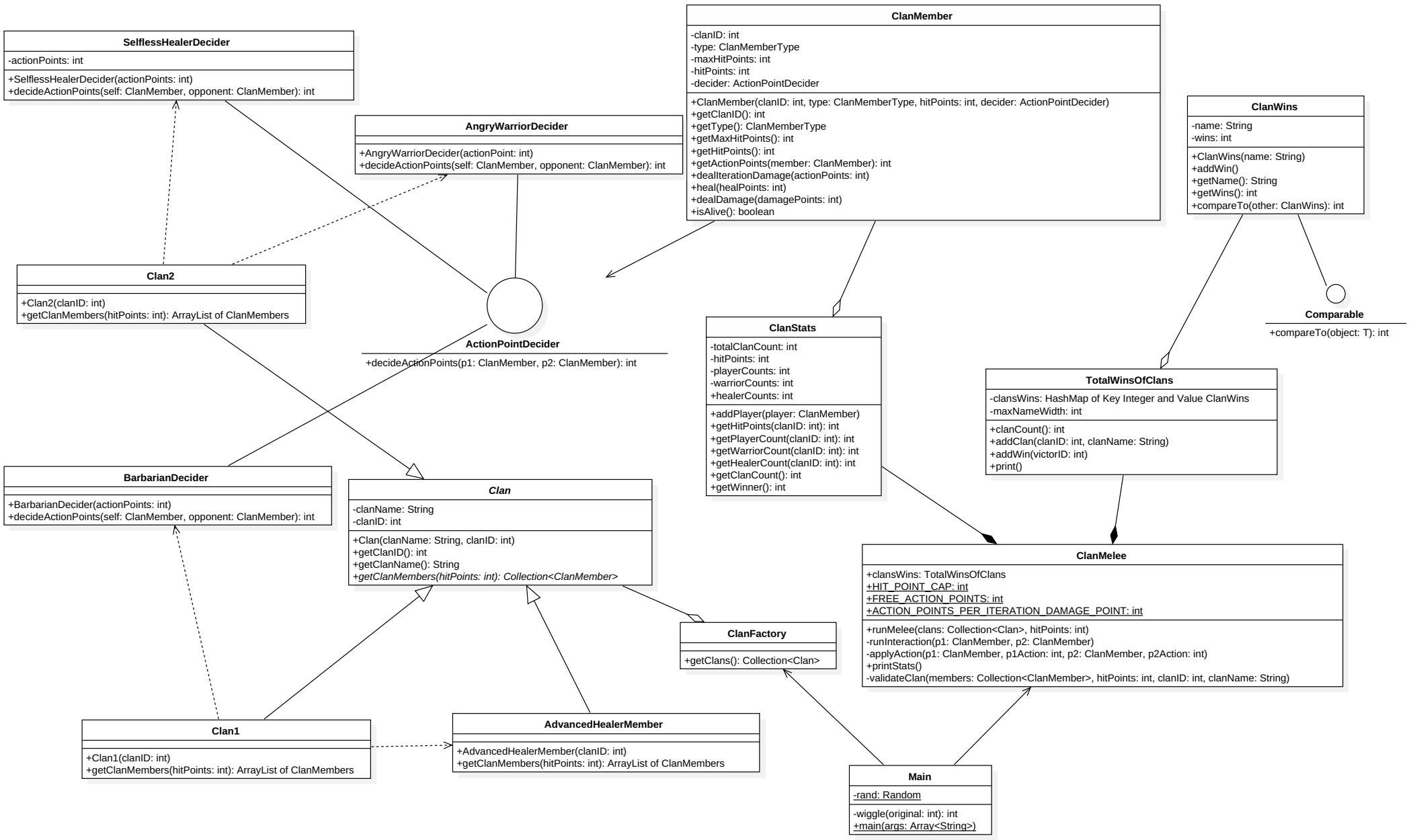
Model::Main


SelflessHealerDecider
-actionPoints: int
+SelflessHealerDecider(actionPoints: int)
+decideActionPoints(self: ClanMember, opponent: ClanMember): int
AngryWarriorDecider
+AngryWarriorDecider(actionPoint: int)
+decideActionPoints(self: ClanMember, opponent: ClanMember): int
ClanMember
-clanID: int
-type: ClanMemberType
-maxHitPoints: int
-hitPoints: int
-decider: ActionPointDecider
+ClanMember(clanID: int, type: ClanMemberType, hitPoints: int, decider: ActionPointDecider)
+getClanID(): int
+getType(): ClanMemberType
+getMaxHitPoints(): int
+getHitPoints(): int
+getActionPoints(member: ClanMember): int
+dealIterationDamage(actionPoints: int)
+heal(healPoints: int)
+dealDamage(damagePoints: int)
+isAlive(): boolean
ClanWins
-name: String
-wins: int
+ClanWins(name: String)
+addWin()
+getName(): String
+getWins(): int
+compareTo(other: ClanWins): int
Clan2
+Clan2(clanID: int)
+getClanMembers(hitPoints: int): ArrayList of ClanMembers
ActionPointDecider
+decideActionPoints(p1: ClanMember, p2: ClanMember): int
ClanStats
-totalClanCount: int
-hitPoints: int
-playerCounts: int
-warriorCounts: int
+healerCounts: int
+addPlayer(player: ClanMember)
+getHitPoints(clanID: int): int
+getPlayerCount(clanID: int): int
+getWarriorCount(clanID: int): int
+getHealerCount(clanID: int): int
+getClanCount(): int
+getWinner(): int
Comparable
+compareTo(object: T): int
TotalWinsOfClans
-clansWins: HashMap of Key Integer and Value ClanWins
-maxNameWidth: int
+clanCount(): int
+addClan(clanID: int, clanName: String)
+addWin(victorID: int)
+print()
BarbarianDecider
+BarbarianDecider(actionPoints: int)
+decideActionPoints(self: ClanMember, opponent: ClanMember): int
Clan
-clanName: String
-clanID: int
+Clan(clanName: String, clanID: int)
+getClanID(): int
+getClanName(): String
+getClanMembers(hitPoints: int): Collection<ClanMember>
ClanMelee
+clansWins: TotalWinsOfClans
+HIT_POINT_CAP: int
+FREE_ACTION_POINTS: int
+ACTION_POINTS_PER_ITERATION_DAMAGE_POINT: int
+runMelee(clans: Collection<Clan>, hitPoints: int)
-runInteraction(p1: ClanMember, p2: ClanMember)
-applyAction(p1: ClanMember, p1Action: int, p2: ClanMember, p2Action: int)
+printStats()
-validateClan(members: Collection<ClanMember>, hitPoints: int, clanID: int, clanName: String)
ClanFactory
+getClans(): Collection<Clan>
Clan1
+Clan1(clanID: int)
+getClanMembers(hitPoints: int): ArrayList of ClanMembers
AdvancedHealerMember
+AdvancedHealerMember(clanID: int)
+getClanMembers(hitPoints: int): ArrayList of ClanMembers
Main
-rand: Random
-wiggle(original: int): int
+main(args: Array<String>)