Monster

-health : int
-age : int
-gender : char
-aggresion : int
-fertility : int
-colour : string
-size : int

+attack() : void
+breed() : void

User

-username : string
-password : string
-money : int
-friends: ArrayList<String>

+setPassword() : void
+getPassword() : string
+getMoney() : int
+addFriend() : User
+deleteFriend() : void
+addMonster(Monster m)
+deleteMonster(Monster m)

-monsters

contains id's of friends

-awaitingRequests :ArrayList

Request

AccountManager

+createAccount(String username, String password)
-removeAccount(String username)
-resetAccount(String username)

Friend

- playerId : String
- wealth : int
- numbOfMonsters : int

<<enum>>
MonsterFilter

ALL,
FOR_SALE,
FOR_BREEDING

DataManager

- playerId : String

+ DataManager(String playerId)
+ getFriends() : ArrayList<Friend>
+ getMonsters(MonsterFilter f) : ArrayList<Monster>
+ getFriendsMonsters(String friendId, MonsterFilter filter)

<>
Request

-sourceId : String
-targetId : String
-waitingForResponse : boolean

+convertToNotification() : String

-type

<<enum>>
RequestType

FRIEND_REQUEST,
ACCEPT_FRIENDSHIP,
DECLINE_FRIENDSHIP,
OFFER_FIGHT,
ACCEPT_FIGHT,
DECLINE_FIGHT,
ACCEPT_BREED_OFFER,
BREEDING_RESOLVED,
BUY_MONSTER;

BreedRequest

-femaleMonsterId : String
-children : ArrayList<Monster>

RequestDispatcher

+RequestDispatcher(String playerId)
+sendFriendRequest(String friendId)
+acceptFriendRequest(FriendRequest request)
+sendFightRequest(String myMonsterId, String friendsMonsterId)
+sendBreedRequest(String myMonsterId, String friendsMonsterId)
+sendBuyRequest(String monsterToBuyId)