CSC Match

# Low Level Design

## CSC205 – Group Project

Emilio Gonzalez

Justin Morris

Adan Partida

Joshua Warren

3/30/2018

# CSCMatch

## CSCMatch()

SET menuInput to NULL

## Main()

NEW CSCMatch()

DO

* showMenu()

WHILE

* menuInput DOES NOT EQUAL x or exit

## showMenu()

DO

* PRINT menu options
* EXECUTE user command

WHILE user input is not “x”

## quit()

EXIT PROGRAM

# Membership

## loadMembers()

GET Members data file input from user

OPEN Members File

SET members equal to loaded data

## saveMembers()

OPEN Members File

* WRITE serializable members list to file

CLOSE Members File

## listMembers()

ITERATE Member in members

* PRINT Member Data
* ITERATE Member Interests
  + PRINT Interest Level and Topic

## addMember()

ITERATE members list

* IF Member IS IN members list
  + UPDATE Member
* ELSE
  + PRINT User input instructions
  + INSTANTIATE new Member
  + ADD Member to members list.

## removeMember(Member)

IF Member is on list

* REMOVE Member from members list

ELSE

* THROW Exception “Member Not Found”

## listMember(Member)

FIND Member in members list

OUTPUT Member data by calling Member.toString()

# Member

## Member()

INSTANTIATE Member with null values

## addInterest(topic: String, level: int)

IF Member hasInterest(topic)

* UPDATE Interest object’s level

ELSE

* CREATE Interest object with topic and level
* ADD Interest to Members interests list.

## removeInterest(topic: String)

IF hasInterest(topic)

* REMOVE Interest from interests list

ELSE

* THROW Exception “Interest Not Found”

## memberCompatability(Member MemberB)

//MemberA is parent object

SET score to 0

FOR EACH of MemberB’s Interests as BInterest

* FOR EACH of MemberA’s Interests as AInterest
  + IF AInterest.topic = BInterest.topic
    - INCREASE score by AInterest.level \* BInterest.level
  + ELSE
    - INCREASE score by BInterest/2

## topMatches(LinkedList<Member> members)

//MemberA is parent object

FOR EACH MemberB in members

* IF MemberA.meid DOES NOT EQUAL MemberB.meid
  + ADD MemberMatch object to matchList where member = MemberB and score = memberCompatability(MemberB)

RETURN first 5 matches in matchList

# MemberMatch

## MemberMatch(member: Member, score: int)

INSTANTIATE new MemberMatch with member and score set as provided.

# OrderedLinkedList<MemberMatch> extends LinkedList<T>

## add(MemberMatch matchObj)

IF list.isEmpty

* ADD matchObj to front of list.

ELSE

* SET inserted = false
* REMOVE matchObj from list
* CREATE iterator for list
* WHILE iterator.next() IS NOT NULL AND inserted=false
  + IF matchObj.score > match.score
    - INSERT matchObj BEFORE match
* IF inserted = false
  + ADD matchObj to end of list

# Interest

## Interest(topic: String, level: int)

INSTANTIATE new Interest Object with topic and level set as provided.