**DEMOGRAPHIC INFORMATION**

1.- Gender: Male ( ) Female (✔)

2.- Age: \_33\_\_\_\_\_\_

3.- Studies: PhD (✔) Master ( ) Bachelor ( ) Student ( ) None ( )

4.- Years of experience in software development: \_\_\_10\_\_\_\_

5.- Years of experience in Java software development: \_\_\_4\_\_\_

6.- Rate from 1 (None) to 5 (Expert) your level of knowledge of Java: 1( ) 2( ) 3( ) 4( ✔) 5( )

**EVALUATION**

**A.** The following Java methods are part of the "ramen" project, which implements a simple social network, and was used in a Java programming course.

1. The method is included in the "MessageDAO" class, and has the following code:

**public** Long \_\_\_\_\_\_\_\_\_\_(Message m) {

**for** (Entry<Long,Message> e : messages.entrySet()) {

**if** (m.equals(e.getValue()))

**return** e.getKey();

}

**return** 0L; // **TODO**: Exception

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| findMessageId | 1( ) 2( ) 3( ) 4( ) 5(✔ ) |  |
| getID | 1( ) 2(✔ ) 3( ) 4( ) 5( ) |  |
| messageDAOMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| getMessageID | 1( ) 2( ) 3( ) 4( ) 5(✔) |  |

2. The method is included in the "Ramen" class, and has the following code:

**public** **boolean** \_\_\_\_\_\_\_\_\_\_(LocalMessage lm) **throws** SQLException {

MessageDAO mdb = *ddb*.getMdb();

mdb.readLocalMessage(currentUser, lm.getReference());

**return** lm.read();

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| readMessage | 1( ) 2( ) 3( ) 4( ) 5(✔) |  |
| readLocalMessage | 1( ) 2( ) 3( ) 4( ) 5(✔ ) |  |
| markMessageAsRead | 1( ) 2( ) 3( ) 4(✔) 5( ) |  |
| ramenMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |

**B.** The following Java methods are part of the "Log4-detector" project, which is a scanner that detects Log4J vulnerabilities.

1. The method is included in the "VersionComparator" class, and has the following code:

**public** **static** **boolean** \_\_\_\_\_\_\_\_\_\_(String s) {

**if** (s == **null** || "".equals(s.trim())) {

**return** **false**;

}

String[] split = *split*(s);

**int** min = Integer.***MAX\_VALUE***;

**for** (**int** i = 0; i < split.length; i++) {

**final** **boolean** isLastWord = (i == split.length - 1);

String tok = split[i];

String[] subSplit = *splitIntoAlphasAndNums*(tok);

min = Math.*min*(min, *minWordScore*(subSplit, isLastWord));

}

**return** min >= 0;

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| isWordScorePositive | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |
| versionComparatorMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| isReleaseVersionStringValid | 1(✔ ) 2( ) 3( ) 4( ) 5( ) |  |
| isReleaseVersion | 1( ) 2(✔) 3( ) 4( ) 5( ) |  |

2. The method is included in the "Strings" class, and has the following code:

**public** **static** **int** \_\_\_\_\_\_\_\_\_\_(String s, **char** c) {

**int** count = 0;

**for** (**char** ch : s.toCharArray()) {

**if** (ch == c) {

count++;

}

}

**return** count;

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| stringMethod | 1(✔ ) 2( ) 3( ) 4( ) 5( ) |  |
| countCharacter | 1( ) 2( ) 3(✔) 4( ) 5( ) |  |
| countChar | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |
| countOccurrences | 1( ) 2( ) 3( ) 4( ) 5(✔ ) |  |

**C.** The following Java methods are part of the "compiler" project, which is a compiler for a simple programming language, for use in education.

1. The method is included in the "SymbolTable" class, and has the following code:

**public** **boolean** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(String lexeme, String type) {

**for** (Symbol symbol : symbols) {

**if** (symbol.lexeme.equals(lexeme)) {

**if** (symbol.kind.toString().equals(type)) {

**return** **true**;

}

}

}

**return** **false**;

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| isLexemeTypeMatch | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |
| symbolTableMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| checkSymbolType | 1( ) 2( ) 3( ) 4(✔) 5( ) |  |
| isSymbolTypeMatching | 1( ) 2( ) 3( ) 4( ) 5(✔ ) |  |

2. The method is included in the "JackClasses" class, and has the following code:

**private** **void** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_() {

dataTypeTransfer.put("char", "(char|int|null)");

dataTypeTransfer.put("String", "(String|null)");

dataTypeTransfer.put("int", "int|Array");

dataTypeTransfer.put("boolean", "(boolean|int)");

dataTypeTransfer.put("Array", "(Array|int|class|null)");

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| jackClassesMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| initializeDataTypeTransferDictionary | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |
| dataTypeTransferDictionary | 1( ) 2( ) 3( ) 4(✔) 5( ) |  |
| populateDataTypeTransferDictionary | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |

**D.** The following Java methods are part of the "jvector" project, which is a pure Java embedded vector search engine, used by DataStax Astra DB and Apache Cassandra.

1. The method is included in the "GraphIndexBuilder" class, and has the following code:

**private** NodeArray \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(SearchResult.NodeScore[] candidates, **int** count, NodeArray scratch) {

scratch.clear();

**for** (**int** i = 0; i < count; i++) {

**var** candidate = candidates[i];

scratch.addInOrder(candidate.node, candidate.score);

}

**return** scratch;

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| addToScratchCandidates | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |
| toScratchCandidates | 1( ) 2(✔) 3( ) 4( ) 5( ) |  |
| graphIndexBuilderMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| sortAndAddCandidates | 1( ) 2( ) 3( ) 4( ) 5(✔ ) |  |

2. The method is included in the "ConcurrentNeighborSet" class, and has the following code:

**public** **void** \_\_\_\_\_\_\_\_\_\_(Function<Integer, ConcurrentNeighborSet> neighborhoodOf, **float** overflow) {

NodeArray neighbors = neighborsRef.get();

**for** (**int** i = 0; i < neighbors.size(); i++) {

**int** nbr = neighbors.node[i];

**float** nbrScore = neighbors.score[i];

ConcurrentNeighborSet nbrNbr = neighborhoodOf.apply(nbr);

nbrNbr.insert(nodeId, nbrScore, overflow);

}

}

For the following method names, indicate whether they are appropriate for the previous method. Rate from 1 (disagree) to 5 (strongly agree) your level of agreement.

|  |  |  |
| --- | --- | --- |
| **Method name** | **Is the name appropriate?** | **Comments (optional)** |
| concurrentNeighborSetMethod | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| createReciprocalLinks | 1(✔) 2( ) 3( ) 4( ) 5( ) |  |
| backlink | 1(✔ ) 2( ) 3( ) 4( ) 5( ) |  |
| addReciprocalLinks | 1( ) 2( ) 3( ) 4(✔ ) 5( ) |  |