**DEMOGRAPHIC INFORMATION**

1.- Gender: Male ( ) Female ( )

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

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

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

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

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 "UserDAO" class, and has the following code:

**public** Long \_\_\_\_\_\_\_\_\_\_\_\_(User u) {

**for** (Map.Entry<Long,User> e : users.entrySet()) {

**if** (u.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)** |
| getUserId | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| findUserId | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| userDAOMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| getID | 1( ) 2( ) 3( ) 4( ) 5( ) |  |

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

**public** **boolean** \_\_\_\_\_\_\_\_\_\_\_(Entity e) **throws** SQLException, ForbiddenAction {

UserDAO udb = *ddb*.getUdb();

**boolean** ret = **true**;

**if**(e **instanceof** Group) {

**for**(Group g : ((Group) e).getSubgroups())

ret = ret & block(g);

}

**return** udb.addBlock(currentUser, e) && ret;

}

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)** |
| blockEntityAndSubgroups | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| ramenMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| block | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| blockEntity | 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 "Bytes" class, and has the following code:

**private** **static** **int**[] \_\_\_\_\_\_\_\_\_\_\_\_(**byte**[] pattern) {

**int**[] failure = **new** **int**[pattern.length];

**int** j = 0;

**for** (**int** i = 1; i < pattern.length; i++) {

**while** (j > 0 && pattern[j] != pattern[i]) {

j = failure[j - 1];

}

**if** (pattern[j] == pattern[i]) {

j++;

}

failure[i] = j;

}

**return** failure;

}

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)** |
| bytesMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| calculateKMPPatternFailure | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| kmpFailure | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| computeFailureArray | 1( ) 2( ) 3( ) 4( ) 5( ) |  |

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

**public** **static** String \_\_\_\_\_\_\_\_\_\_\_\_\_(Object o) {

**if** (o == **null**) {

**return** "";

} **else** **if** (o **instanceof** String) {

**return** ((String) o).trim();

} **else** {

**return** o.toString().trim();

}

}

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)** |
| trimObject | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| stringsMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| trimNullSafe | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| nullSafeTrim | 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) {

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

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

varName = symbol.lexeme;

memorySegment = symbol.memorySegment;

offset = symbol.offset;

dataType = symbol.assignDataType;

initOrNot = symbol.initOrNot;

**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)** |
| findIdentifierSymbol | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| isIdentifierSymbolPresent | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| findSymbol | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| symbolTableCompilerMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |

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

**public** **boolean** \_\_\_\_\_\_\_\_\_\_\_\_\_\_(String methodName) {

**for** (String method : methods) {

**if** (method.equals(methodName)) {

**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)** |
| isMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| methodMatch | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| jackClassesMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| isMethodPresent | 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 "SimpleMappedReader" class, and has the following code:

**public** **void** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_() {

**if** (***unsafe*** != **null**) {

**try** {

***unsafe***.invokeCleaner(mbb);

} **catch** (IllegalArgumentException e) {

// empty catch, this was a duplicated/indirect buffer or

// otherwise not cleanable

}

}

}

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)** |
| closeBuffer | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| cleanUnsafeDirectBuffer | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| close | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| simpleMappedReaderMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |

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

**public** **static** <T> Map<Integer, Integer> \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(GraphIndex<T> graph) {

**try** (**var** view = graph.getView()) {

Map<Integer, Integer> oldToNewMap = **new** HashMap<>();

**int** nextOrdinal = 0;

**for** (**int** i = 0; i < view.getIdUpperBound(); i++) {

**if** (graph.containsNode(i)) {

oldToNewMap.put(i, nextOrdinal++);

}

}

**return** oldToNewMap;

} **catch** (Exception e) {

**throw** **new** RuntimeException(e);

}

}

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)** |
| getSequentialOrdinalMapping | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| assignNewOrdinals | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| onDiskGraphIndexMethod | 1( ) 2( ) 3( ) 4( ) 5( ) |  |
| getSequentialRenumbering | 1( ) 2( ) 3( ) 4( ) 5( ) |  |