Abbreviations in identifiers

Thank you for wanting to participate in our survey of the usage of identifier abbreviations in a code visualization tool (CVT).

If you are unfamiliar with a CVT or want a brief description for the CVT used in this survey called Doxygen use this link: Doxygen documentation

The questions are based on the CVT and each section contains a link to the relevant CVT, multiple sections also use the same repository for the questions.

The questionnaire is divided into four parts.

The first is this one, with some pre-questions about you and your knowledge.

The second is three sections about the same project.

The third is three sections about another project.

In the headers for both of the second and third parts, the same information will be displayed. There will be a link to doxygen, and some information about the projects purpose.

Lastly there are some post-questions about the questionnaire itself.

One part of the questions uses a CVT with an abbreviated resolver integrated and the other part don´t.

The email address is only stored temporarily, and any personal information will be removed before data-handling is done.

Thank you for your participation.

Anton & Sven

* Anger obligatorisk fråga

1. Have you read the above introductory text and checked out the Doxygen documentation? *

   Markera endast en oval.

   ○ Yes

Pre-questions

2. 2. What is your current working title? *

Markera endast en oval.

○ Programmer
○ Software Designer
○ Software Architect
○ Software Engineer
○ Architect of other types
○ Frontend Developer
○ Backend Developer
○ Fullstack Developer
○ Övrigt: ____________________

3. 3. How do you rate your expertise in programming? *

Markera endast en oval.

○ Very low
○ Low
○ Average
○ High
○ Very high

4. 4. How many years of programming experience do you have? *

For example 2 years, 7 years, etc.

____

5. 5. How do you rate your expertise in programming with Java? *

Markera endast en oval.

- ○ Very low
- ○ Low
- ○ Average
- ○ High
- ○ Very high

6. 6. How many years of programming experience do you have with Java? *

For example 2 years, 7 years, etc.

____

7. 7. How familiar are you with Code Visualization Tools? *

Markera endast en oval.

- ○ Very Unfamiliar
- ○ Unfamiliar
- ○ Somewhat Familiar
- ○ Familiar
- ○ Very Familiar

Abbreviations in identifiers:

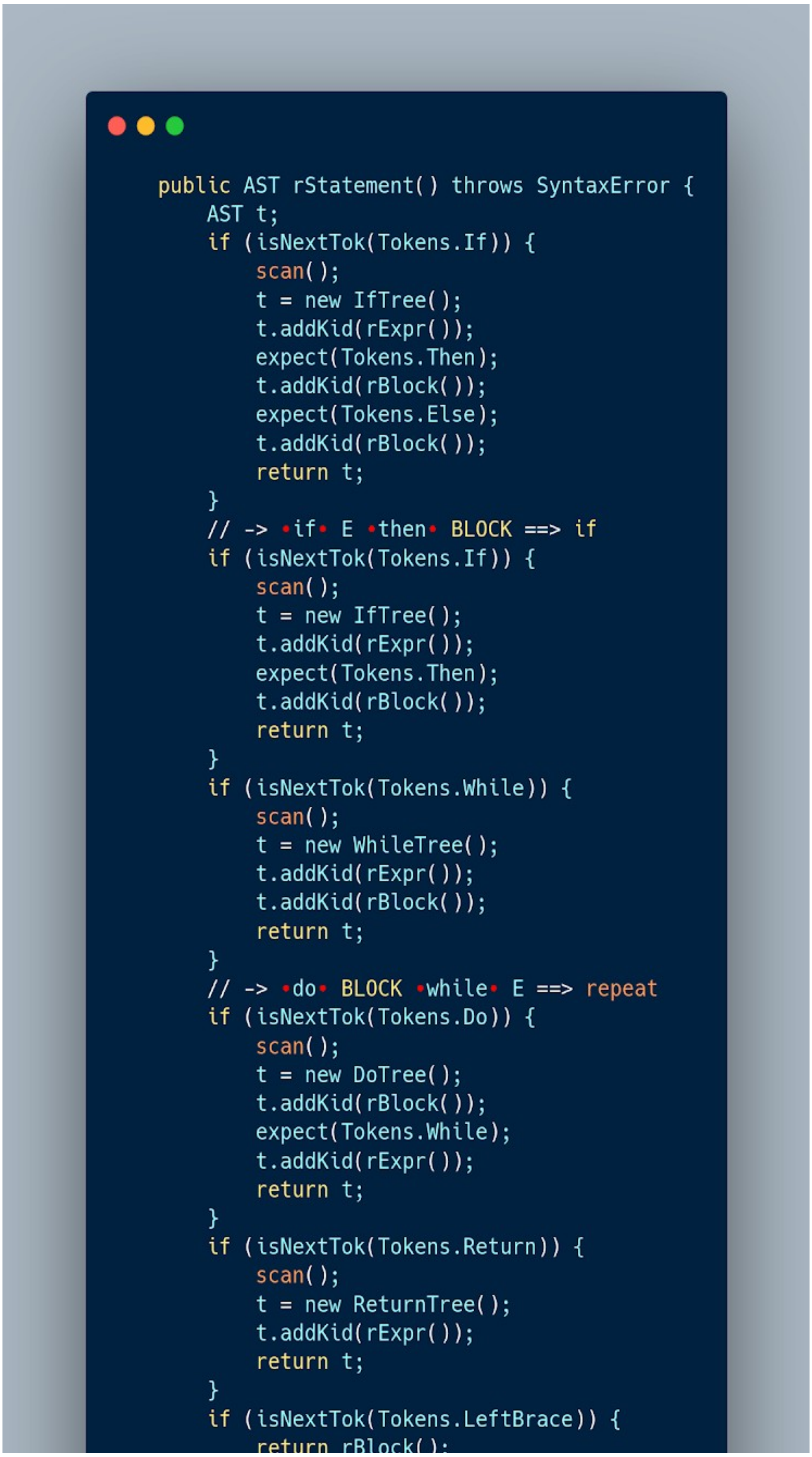
Doxygen (CVT) link: Doxygen
Use the link and try to answer the questions about the abbreviations and the program flow.

Information about the project:
The project is a lexer and parser for a self-made programming-language, with programs looking like this:

```
program { int i int j
  i = i + j + 7
  j = write(i)
}
```

Questions below will be based on the contents of the link that displays the program in a code visualization tool.

8. 1. What is the unabbreviated name of the return-type for this method? *



```
public AST rStatement() throws SyntaxError {
    AST t;
    if (isNextTok(Tokens.If)) {
        scan();
        t = new IfTree();
        t.addKid(rExpr());
        expect(Tokens.Then);
        t.addKid(rBlock());
        expect(Tokens.Else);
        t.addKid(rBlock());
        return t;
    }
    // -> •if• E •then• BLOCK ==> if
    if (isNextTok(Tokens.If)) {
        scan();
        t = new IfTree();
        t.addKid(rExpr());
        expect(Tokens.Then);
        t.addKid(rBlock());
        return t;
    }
    if (isNextTok(Tokens.While)) {
        scan();
        t = new WhileTree();
        t.addKid(rExpr());
        t.addKid(rBlock());
        return t;
    }
    // -> •do• BLOCK •while• E ==> repeat
    if (isNextTok(Tokens.Do)) {
        scan();
        t = new DoTree();
        t.addKid(rBlock());
        expect(Tokens.While);
        t.addKid(rExpr());
        return t;
    }
    if (isNextTok(Tokens.Return)) {
        scan();
        t = new ReturnTree();
        t.addKid(rExpr());
        return t;
    }
    if (isNextTok(Tokens.LeftBrace)) {
        return rBlock();
```

9. 2. In the above image, what does 'isNextTok' check? *

10. 3. How likely is it you understood the abbreviations? *

Markera endast en oval.

○ Very Unlikely
○ Unlikely
○ Neutral
○ Likely
○ Very Likely

Abbreviations in identifiers:

Doxygen (CVT) link: Doxygen
Use the link and try to answer the questions about the abbreviations and the program flow.

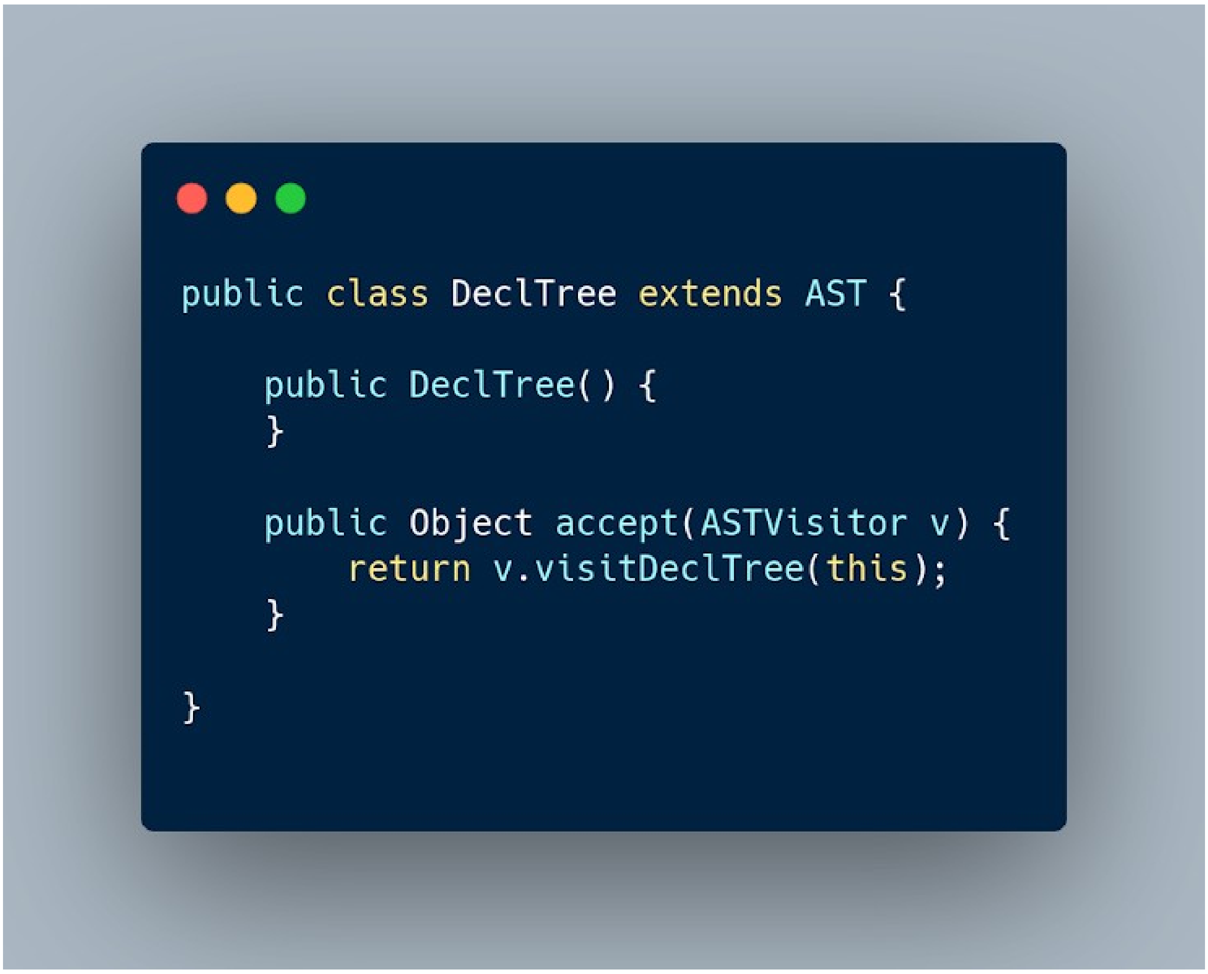
Information about the project:
The project is a lexer and parser for a self-made programming-language, with programs looking like this:

```
program { int i int j
  i = i + j + 7
  j = write(i)
}
```

Questions below will be based on the contents of the link that displays the program in a code visulization tool.

11. 1. What is the use-case for this class? *

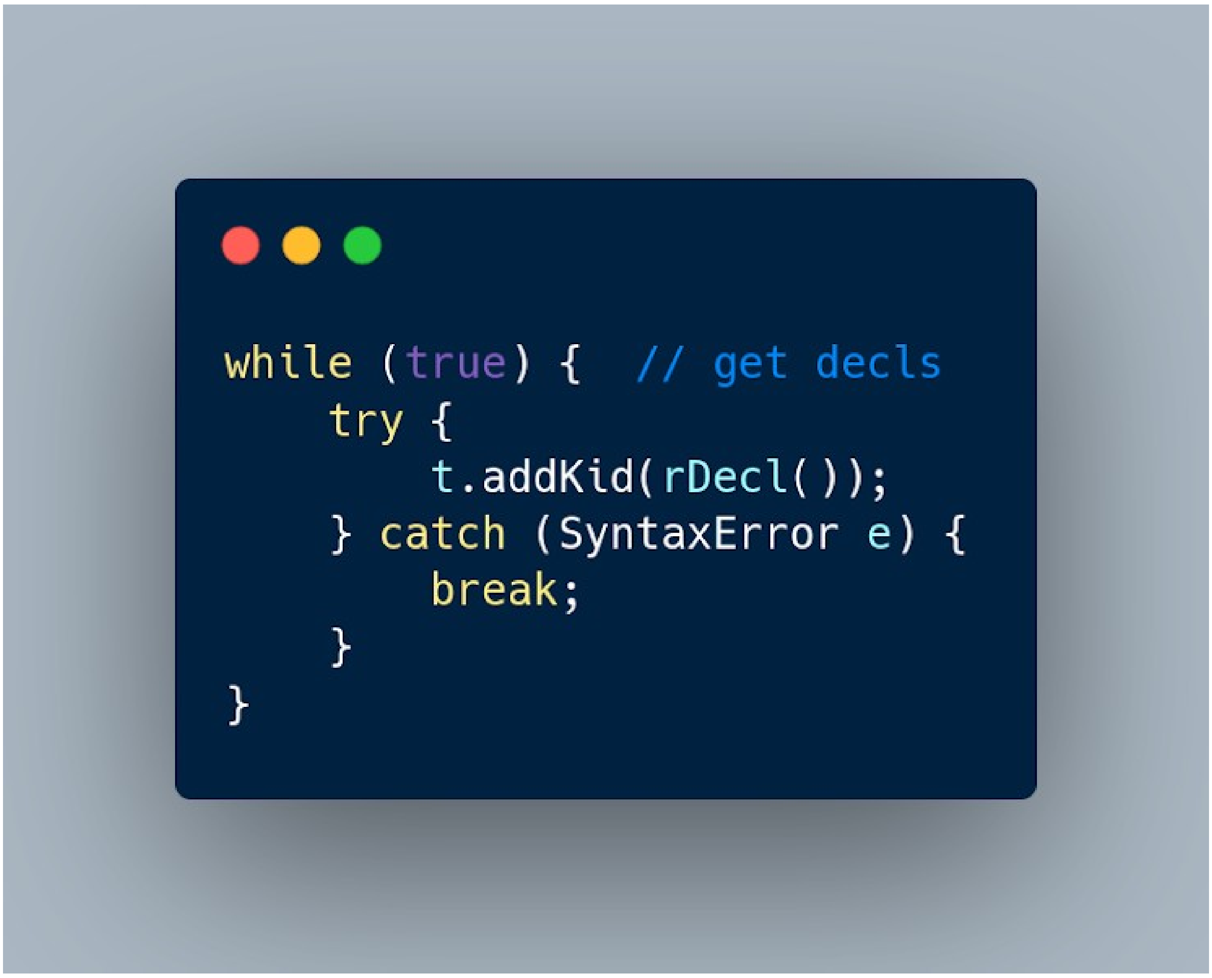
For example, is it used for indicating knowledge of something?



```java
public class DeclTree extends AST {

    public DeclTree() {
    }

    public Object accept(ASTVisitor v) {
        return v.visitDeclTree(this);
    }

}
```

12. 2. In the above image, what is the full name of DeclTree? *

13. 3. What is the unabbreviated name of rDecl()? *



```
while (true) {  // get decls
    try {
        t.addKid(rDecl());
    } catch (SyntaxError e) {
        break;
    }
}
```

14. 4. How likely is it you understood the abbreviations? *

Markera endast en oval.

○ Very Unlikely

○ Unlikely

○ Neutral

○ Likely

○ Very Likely

Abbreviations in identifiers:

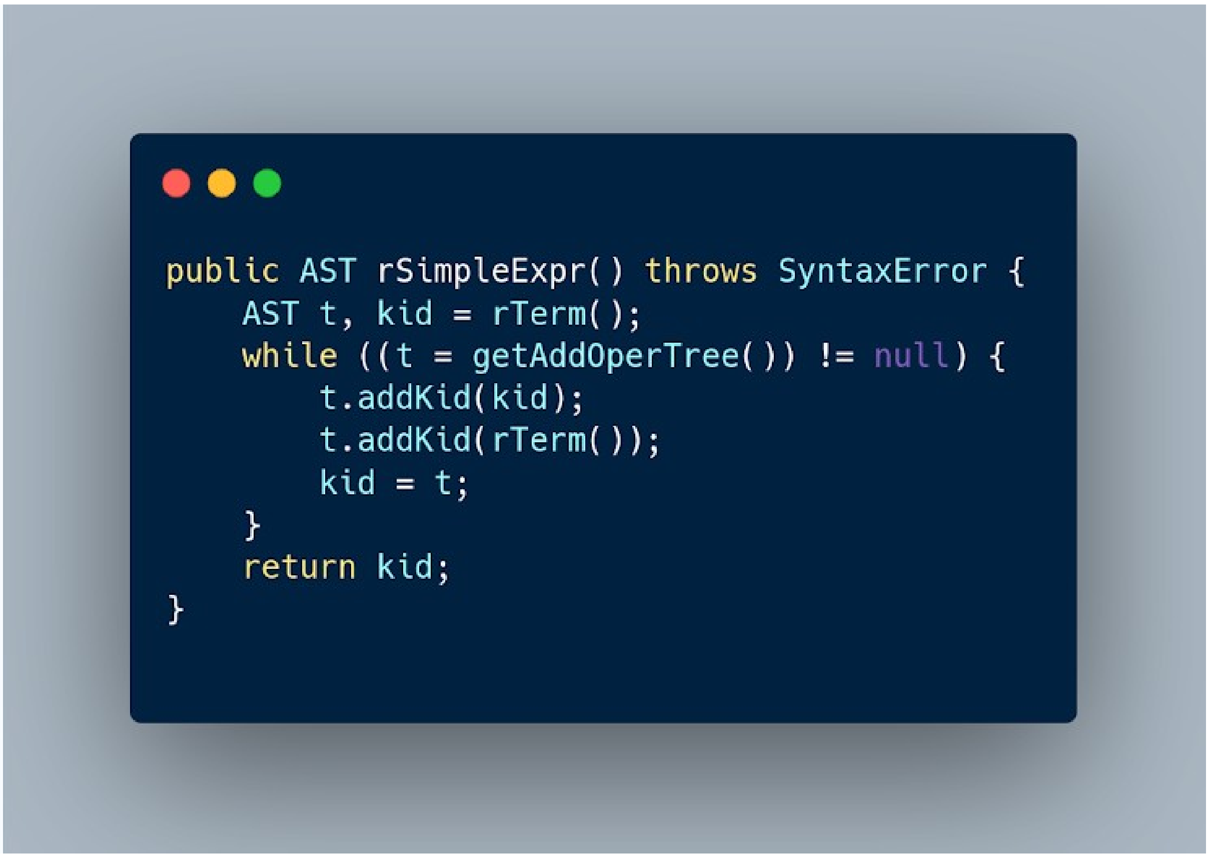
Doxygen (CVT) link: Doxygen
Use the link and try to answer the questions about the abbreviations and the program flow.

Information about the project:
The project is a lexer and parser for a self-made programming-language, with programs looking like this:

```
program { int i int j
   i = i + j + 7
   j = write(i)
}
```

Questions below will be based on the contents of the link that displays the program in a code visulization tool.

15. 1. The method nextToken() is called from rSimpleExpr(), check the boxes that makes up the shortest call-path in the call-tree from rSimpleExpr() to nextToken() *



```
public AST rSimpleExpr() throws SyntaxError {
    AST t, kid = rTerm();
    while ((t = getAddOperTree()) != null) {
        t.addKid(kid);
        t.addKid(rTerm());
        kid = t;
    }
    return kid;
}
```

Markera alla som gäller.

□ rTerm
□ addKid
□ getAddOperTree
□ rFactor
□ expect
□ rName
□ rExpr
□ Scan
□ isNextTok
□ nextToken
□ getMultOperTree

16. 2. In the above method one of the call-paths goes through getMultOperTree, what is it´s unabbreviated name? *

17. 3. How likely is it you understood the abbreviations? *

Markera endast en oval.

○ Very Unlikely

○ Unlikely

○ Neutral

○ Likely

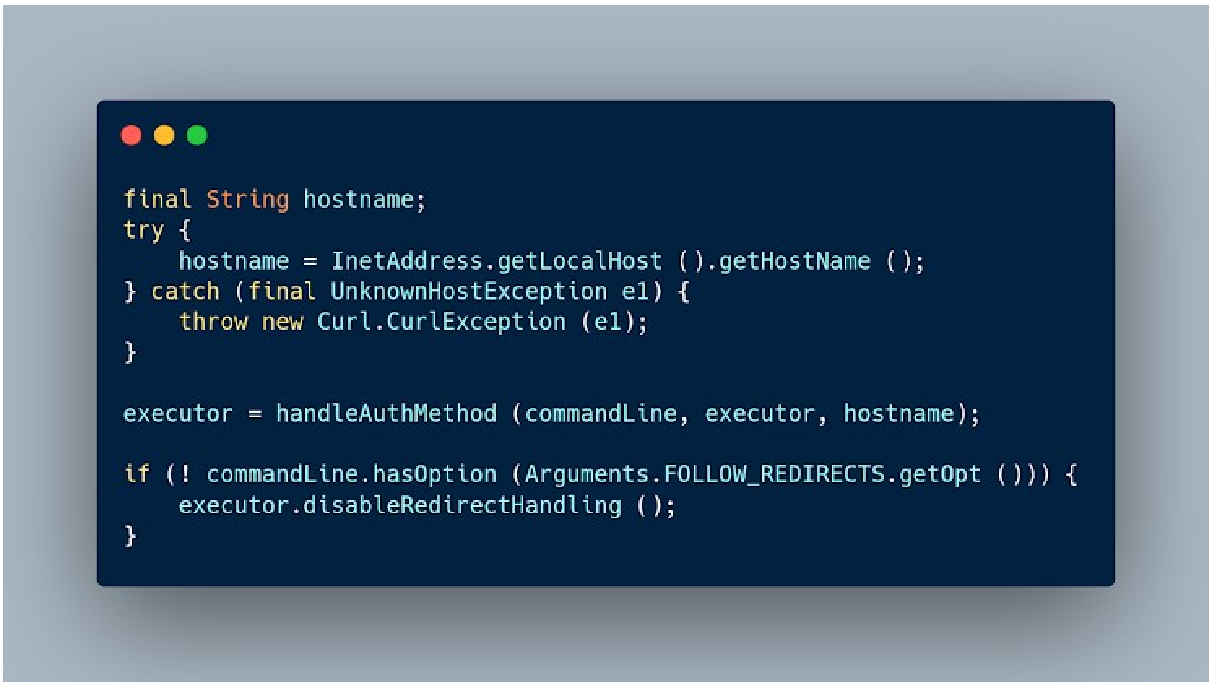
○ Very Likely

cURL

Doxygen (CVT) link: Doxygen
Use the link and try to answer the questions about the abbreviations and the program �ow.

The tool is a cURL-tool written in Java.

18. 1. What does ´Auth´ stand for, in handleAuthMethod? *



```
final String hostname;
try {
    hostname = InetAddress.getLocalHost ().getHostName ();
} catch (final UnknownHostException e1) {
    throw new Curl.CurlException (e1);
}

executor = handleAuthMethod (commandLine, executor, hostname);

if (! commandLine.hasOption (Arguments.FOLLOW_REDIRECTS.getOpt ())) {
    executor.disableRedirectHandling ();
}
```

____

19. 2. How likely is it you understood the abbreviations? *

Markera endast en oval.

○ Very Unlikely

○ Unlikely

○ Neutral

○ Likely

○ Very Likely

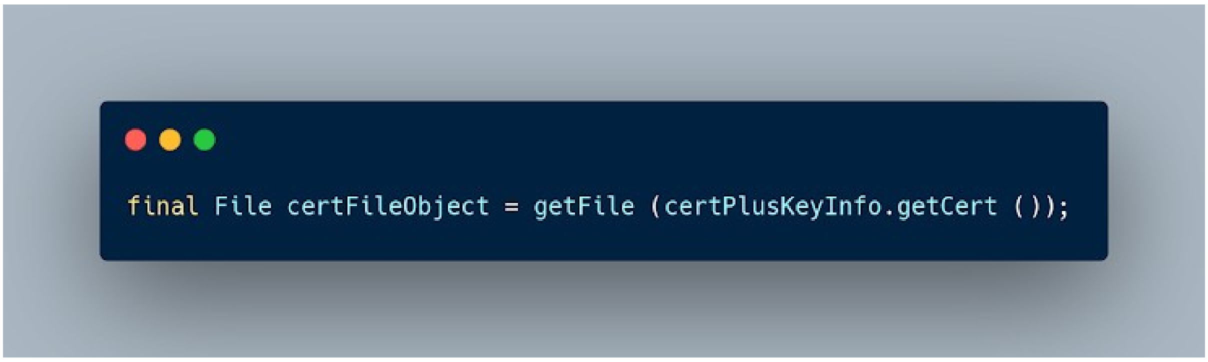
cURL

Doxygen (CVT) link: Doxygen
Use the link and try to answer the questions about the abbreviations and the program flow.

The tool is a cURL-tool written in Java.

20. 1. In the method generateKeyStore() one can find the line in the image. *

What does the Cert in getCert mean?



```
final File certFileObject = getFile (certPlusKeyInfo.getCert ());
```

21. 2. How likely is it you understood the abbreviations? *

Markera endast en oval.

○ Very Unlikely

○ Unlikely

○ Neutral

○ Likely

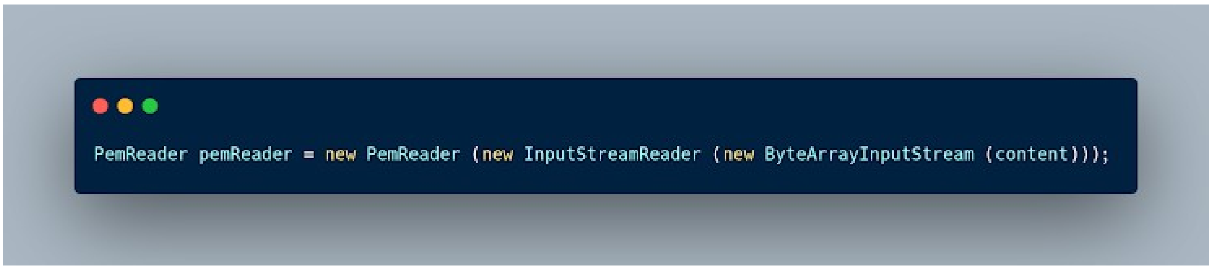
○ Very Likely

cURL

Doxygen (CVT) link: Doxygen
Use the link and try to answer the questions about the abbreviations and the program flow.

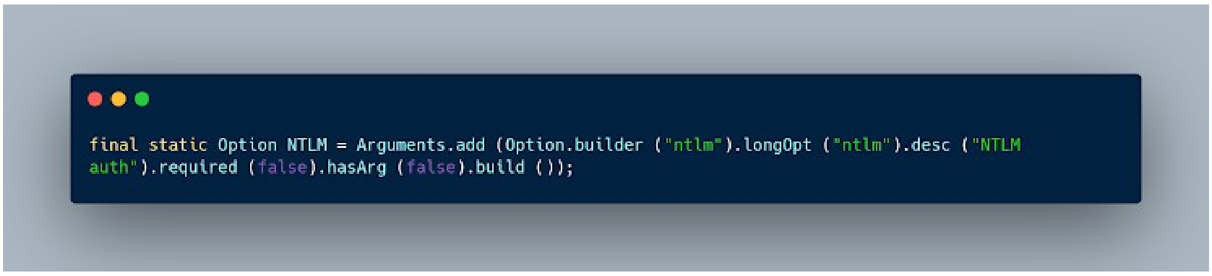
The tool is a cURL-tool written in Java.

22. 1. What is the unabbreviated name for PemReader? *



```java
PemReader pemReader = new PemReader (new InputStreamReader (new ByteArrayInputStream (content)));
```

23. 2. What is the unabbreviated name for NTLM? *



```java
final static Option NTLM = Arguments.add (Option.builder ("ntlm").longOpt ("ntlm").desc ("NTLM
auth").required (false).hasArg (false).build ());
```

24. 3. How likely is it you understood the abbreviations? *

Markera endast en oval.

○ Very Unlikely

○ Unlikely

○ Neutral

○ Likely

○ Very Likely

End of survey

Overall questions about your experience and thoughts.

25. 1.How clear were the task objectives? *

Markera endast en oval.

○ Very unclear

○ Slightly unclear

○ Neither unclear nor clear

○ Slightly clear

○ Very clear

26. 2. Do you think that having the full-named identifiers helped you in code comprehension more than the abbreviated identifiers. *

Markera endast en oval.

○ Full-named identifiers made code comprehension a lot more complicated for me

○ Full-named identifiers made code comprehension a bit more complicated for me

○ Full-named identifiers did not help or hinder me in code comprehension

○ Full-named identifiers helped me a bit in code comprehension

○ Full-named identifiers helped me a lot in code comprehension

27. 3. Any reason for your answer to the above question? *

____

____

____

____

____

28. 4. Do you think that the Abbreviation Resolver helped you in code comprehension? *

Markera endast en oval.

○ Abbreviation Resolver made code comprehension a lot more complicated for me

○ Abbreviation Resolver made code comprehension a bit more complicated for me

○ Abbreviation Resolver did not help or hinder me in code comprehension

○ Abbreviation Resolver helped me a bit in code comprehension

○ Abbreviation Resolver helped me a lot in code comprehension

29. 5. Any reason for your answer to the above question? *

____

____

____

____

____

30. 6. The abbreviation resolver in the Code Visualization Tool show the *full meaning of the abbreviations in a clear and understandable way.

Markera endast en oval.

○ Strongly Disagree
○ Disagree
○ Neither agree or disagree
○ Agree
○ Strongly Agree

31. 7. The Abbreviation Resolver in the Code Visualization Tool save you *time in understanding the abbreviation.

Markera endast en oval.

○ Strongly Disagree
○ Disagree
○ Neither agree or disagree
○ Agree
○ Strongly Agree

32. 8. Any last suggestions or comments

Det här innehållet har varken skapats eller godkänts av Google.

Google Formulär