

Lab03 of CSE244: Java program documentation

1. General Aims

This lab is designed to review basic Java programming skills and introduce the importance of proper commenting when writing code. Javadoc (originally named JavaDoc) is a documentation generator created by Sun Microsystems for the Java language (now owned by Oracle Corporation) for generating API documentation in HTML format from Java source code. The HTML format is used to hyperlink related documents together. To be able to use Javadoc, you have to comment your Java code properly. We will learn how to do so in this lab. With the documentation generated, one will not need to read Java source code to know how to use the Java class. The following is an example document of Math class in Java:

The screenshot shows the Java Platform SE 7 documentation page for the `Math` class. The page is titled "Math (Java Platform SE 7)" and the URL is <https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>. The page includes a navigation bar with links like "Overview", "Package", "Class", "Use", "Tree", "Deprecated", "Index", and "Help". The main content area shows the class hierarchy: `java.lang.Object` and `java.lang.Math`. The class is defined as `public final class Math extends Object`. The page also includes a "Field Summary" section with a table of fields and a "Method Summary" section with a table of methods.

Field	Field and Description
<code>E</code>	The double value that is closer than any other to e , the base of the natural logarithms.
<code>PI</code>	The double value that is closer than any other to π , the ratio of the circumference of a circle to its diameter.

Method	Method and Description
<code>abs(double a)</code>	Returns the absolute value of a double value.
<code>abs(float a)</code>	Returns the absolute value of a float value.
<code>abs(int a)</code>	Returns the absolute value of an int value.
<code>abs(long a)</code>	Returns the absolute value of a long value.
<code>acos(double a)</code>	Returns the arc cosine of a value; the returned angle is in the range 0.0 through π .
<code>asin(double a)</code>	Returns the arc sine of a value; the returned angle is in the range $-\pi/2$ through $\pi/2$.

2. Tasks

Tasks 1 to 3 are designed for you to learn how to use documentation generator (**javadoc**), which parses declarations and documentation comments in a set of Java source files and produces a set of HTML pages describing the public and protected classes, interfaces, constructors, methods and fields.

[Task 1] Read the followings: “Proper Use of Commenting in Java”.

Proper Use of Commenting in Java

Introduction

Proper commenting when writing code is almost as important as the code itself. Analyzing, understanding and adapting code that has been written by others is paramount to the concept of modularization. Writing code from scratch requires a lot of work. If the code is not commented properly it is very difficult to reuse. Without proper commenting a program is just a lot of spaghetti code. For each reuse the spaghetti has to be rediscovered to decide how it can be reused.

With proper commenting, all classes and methods with their associated parameters and return types are described making comprehension straight forward.

Sun has developed a solution. Java has a tool called `Javadoc` that makes the job of creating the documentation easy, **if the commenting has been done properly**. `Javadoc` creates linked html files that describe all the key components of a Java program based on the commenting in the code itself. Therefore when the code is changed, new documentation can be generated to reflect the changes with very little effort. This is why proper commenting is so important when writing code.

Syntax

The syntax of java commenting that allows `Javadoc` to generate documentation is relatively simple. Commenting that starts with a `/**` and ends with a `*/` will be converted by `Javadoc`. Comment should be placed immediately before the element. The three elements that can be commented are: **class**, **method** or **variable**. `Javadoc` will only create docs for `public` and `protected` elements; it will not create docs for `private` nor `friendly` elements.

Here is a simple example:

```
/** This is an example of a class comment,
 *  comments can also be multiple lines, leading """
 *  will be ignored
 *
 * @author Yawei Liang
 * @version Sept 21, 2017
 */
public class Simple
{
    /** This is an example of a variable comment,
     *  comments for variables within methods
     *  will not be documented
     */
    public int i;
```

```
/** This is an example of a method comment
 * @param value of an int arguments
 * @return is a void return type so there will be no return
 */
public void method (int value)
{
}
}
```

Check `index.html` in your `SimpleClassJavaDoc` folder from dropbox at “\CSE244\Labs\Lab03-JavaDoc” to see the documentation.

There are several “doc tags” that can be used within the comment that will generate special paragraphs in the documentation. Doc tags start with the @ symbol. There are class doc tags and method doc tags.

Class doc tags

`@version` comment. Creates a paragraph about the version of the Class.

`@author` comment. Creates a paragraph about the author of the Class. There can be multiple author tags but they must be consecutive.

Method doc tags

`@param` name comments. Creates a paragraph about the parameter named. It will include all comments following the name until another doc tag is found.

`@return` comment. Creates a paragraph about the return value of the method.

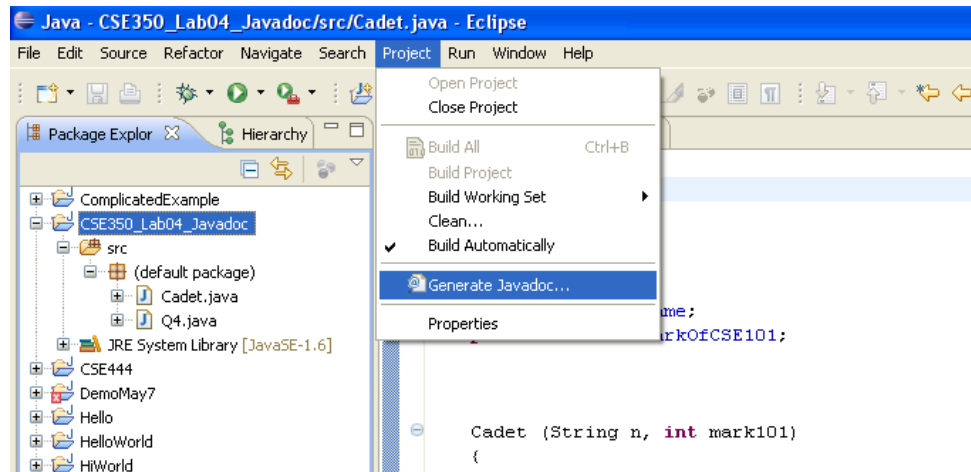
For more information about Javadoc commenting, please see the webpage at:

<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

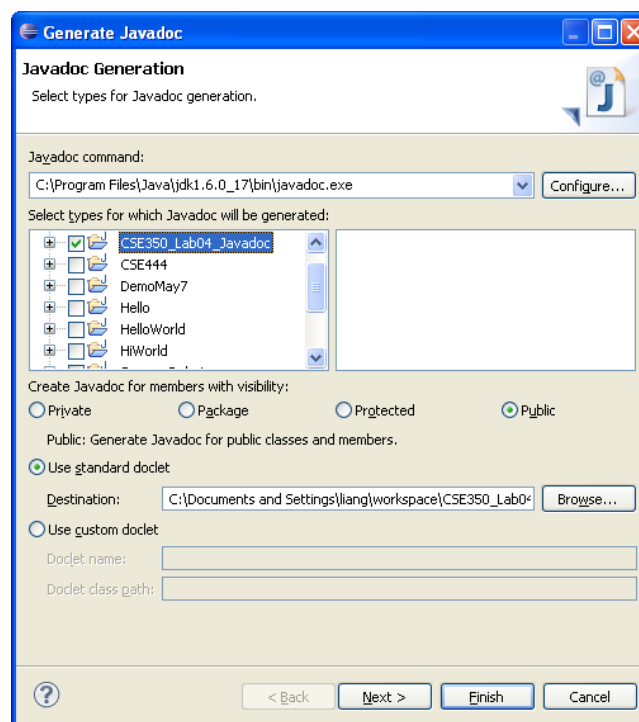
[Task 2] Reading the followings: “Generating Javadoc in Eclipse IDE” and generate Javadoc for the Java code commented in Task 1.

Generating Javadoc in Eclipse IDE

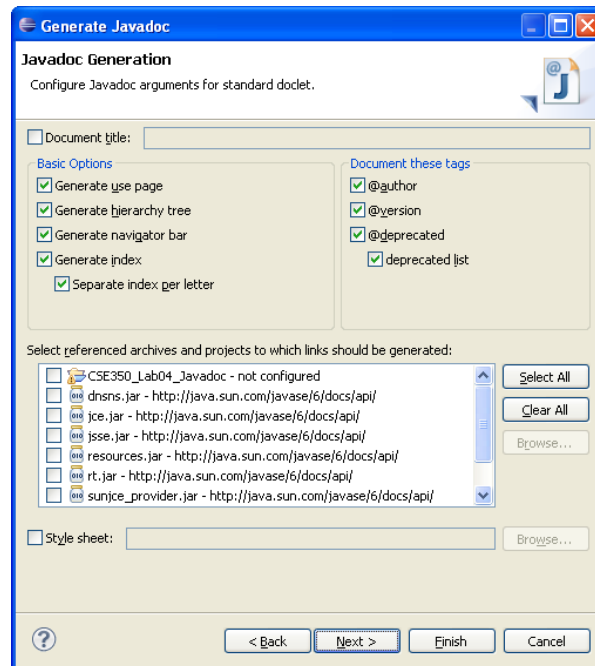
1. Create your project; import those classes (Java files) associated to the project.
2. Prepared comments in your classes by following a Javadoc comment style.
3. Click on Project link and choose “Generate Javadoc” option



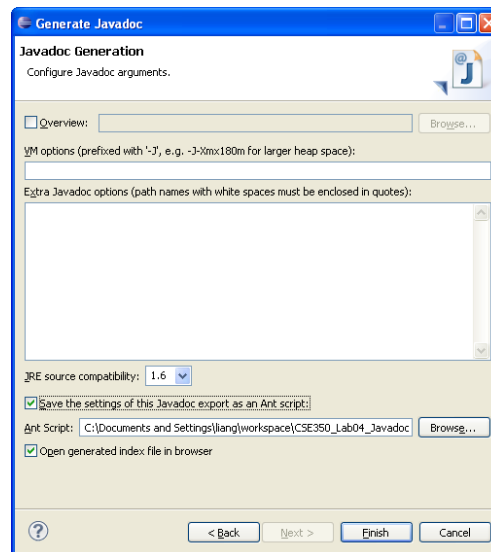
- Now a window, “Generate Javadoc”, will be opened where you can select Java Projects or their underlying resources for which Javadoc needs to be generated. Several other options are also there where user can select any of them as per the need. Here user can select whether to generate Javadoc for public/private API’s etc.



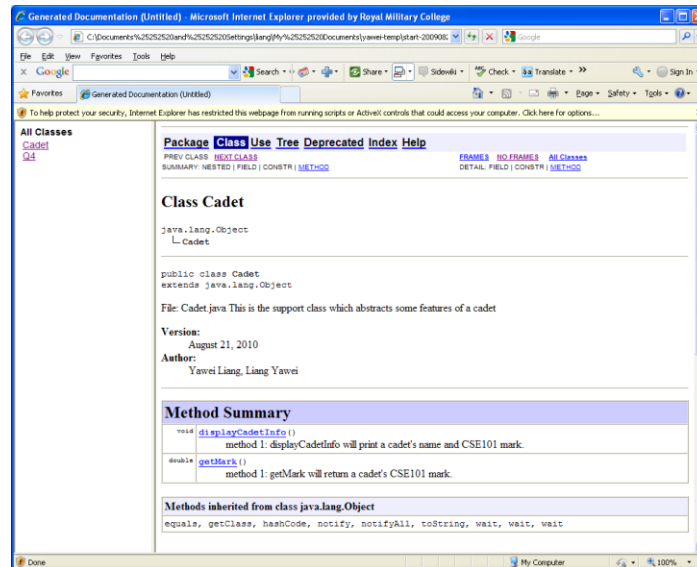
- Now IDE asks for other features as well for the generation of Javadoc. A Java programmer should be familiar with these options.



6. You can also save settings as ant script so that you can use the script to generate javadoc in future.



7. Click “Finish” Javadoc will be generated. If you select option of opening index file in browser then after generation of Javadoc you will find ” index.htm ” of Javadoc in your default Web Browser. On console you can see progress of JavaDoc Generation.



[Task 3] Now, go back to your Lab02 Java code, and edit the Java files to include proper commenting, be verbose. Comment each class, variable and method. Comments should have a brief description of what the code does and how it does it. And use the skills you learnt from Tasks 1 and 2 to generate documentations for those two Java programs. You must submit your commented Java code only from this task.

3. Deliverables and Deadline

To Dr. Yawei Liang, you should only email the Java files commented in Task 3 to liang-y@rmc.ca. The deadline for this lab is at 5:00pm, Oct 5, 2018.

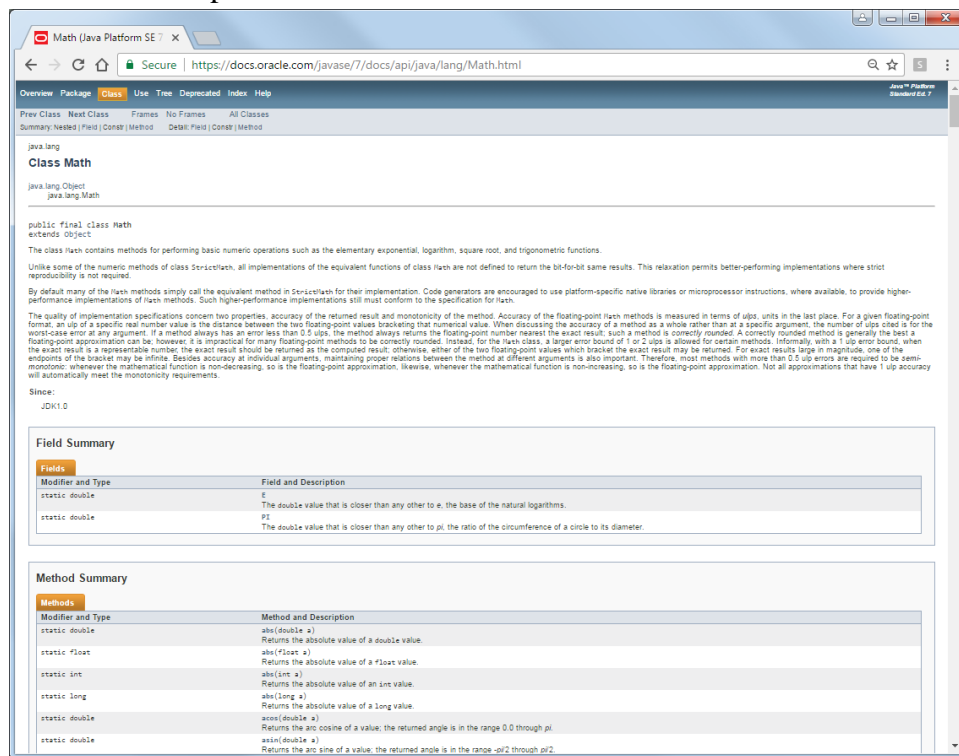
4. Marking Scheme

Task Description		Marks	Check (instructor)
1	Proper commenting of classes, variables and methods.		
	Classes (author, version, content description)	4	
	Variables, and	3	
	Methods.	3	
Total:		10	

Lab03 de INF244: Documentation du programme Java

1. Objectifs généraux

Ce laboratoire est conçu pour passer en revue les compétences de base en programmation Java et introduire l'importance de commenter correctement lors de la rédaction du code. Javadoc (initialement nommé JavaDoc) est un générateur de documentation créé par Sun Microsystems pour le langage Java (maintenant détenu par Oracle Corporation) pour générer la documentation de l'API au format HTML à partir du code source Java. Le format HTML est utilisé pour lier des documents liés entre eux. Pour pouvoir utiliser Javadoc, vous devez commenter correctement votre code Java. Nous allons apprendre à le faire dans ce laboratoire. Avec la documentation générée, il ne sera pas nécessaire de lire le code source Java pour savoir comment utiliser la classe Java. Voici un exemple de document de classe Math en Java :



2. Tâches

Les tâches 1 à 3 sont conçues pour apprendre à utiliser le générateur de documentation (javadoc), qui analyse les déclarations et les commentaires de documentation dans un ensemble de fichiers source Java et produit un ensemble de pages HTML décrivant les classes, interfaces, constructeurs, méthodes et champs.

[**Tâche 1**] Lisez les éléments suivants: «Utilisation appropriée des commentaires en Java».

Utilisation correcte des commentaires en Java

Introduction

Les commentaires appropriés lors de l'écriture du code sont presque aussi importants que le code lui-même. Analyser, comprendre et adapter le code qui a été écrit par d'autres est primordial pour le concept de modularisation. Ecrire du code à partir de zéro nécessite beaucoup de travail. Si le code n'est pas correctement commenté, il est très difficile de le réutiliser. Sans commentaires appropriés, un programme ne contient que du code spaghetti. Pour chaque réutilisation, les spaghettis doivent être redécouverts pour décider de la manière dont ils peuvent être réutilisés.

Avec des commentaires corrects, toutes les classes et méthodes avec leurs paramètres et types de retour associés sont décrites, ce qui facilite la compréhension.

Sun a développé une solution. Java a un outil appelé Javadoc qui facilite la création de la documentation, si les commentaires ont été faits correctement. Javadoc crée des fichiers HTML liés qui décrivent tous les composants clés d'un programme Java en fonction des commentaires dans le code lui-même. Par conséquent, lorsque le code est modifié, une nouvelle documentation peut être générée pour refléter les modifications avec très peu d'effort. C'est pourquoi un commentaire correct est si important lors de l'écriture de code.

Syntaxe

La syntaxe des commentaires Java permettant à Javadoc de générer de la documentation est relativement simple. Commentaire qui commence par `a /**` et se termine par un `*/` avec être converti par Javadoc. Le commentaire doit être placé immédiatement avant l'élément. Les trois éléments pouvant être commentés sont: class, method ou variable. Javadoc ne créera que des documents pour les éléments publics et protégés; il ne créera pas de documents pour des éléments privés ni amicaux.

Voici un exemple simple:

```
/** This is an example of a class comment,
 *  comments can also be multiple lines, leading ""
 *  will be ignored
 *
 * @author Yawei Liang
 * @version Sept 21, 2017
 */
public class Simple
{
    /** This is an example of a variable comment,
     *  comments for variables within methods
```



```
*    will not be documented
*/
public int i;

/** This is an example of a method comment
 * @param value of an int arguments
 * @return is a void return type so there will be no return
 */
public void method (int value)
{
}
}
```

Vérifiez index.html dans votre dossier SimpleClassJavaDoc depuis la liste déroulante «\ CSE244 \ Labs \ Lab03-JavaDoc» pour voir la documentation.

Il existe plusieurs «balises doc» pouvant être utilisées dans le commentaire pour générer des paragraphes spéciaux dans la documentation. Les balises Doc commencent par le symbole @. Il existe des balises de classe et des balises de méthode.

Balises doc de classe

Commentaire @version crée un paragraphe sur la version de la classe.

Commentaire de @author crée un paragraphe sur l'auteur de la classe. Il peut y avoir plusieurs balises auteur mais elles doivent être consécutives.

Mots-clés de la méthode

@param nom commentaires. Crée un paragraphe sur le paramètre nommé. Il inclura tous les commentaires suivant le nom jusqu'à ce qu'une autre balise doc soit trouvée.

Commentaire de retour crée un paragraphe sur la valeur de retour de la méthode.

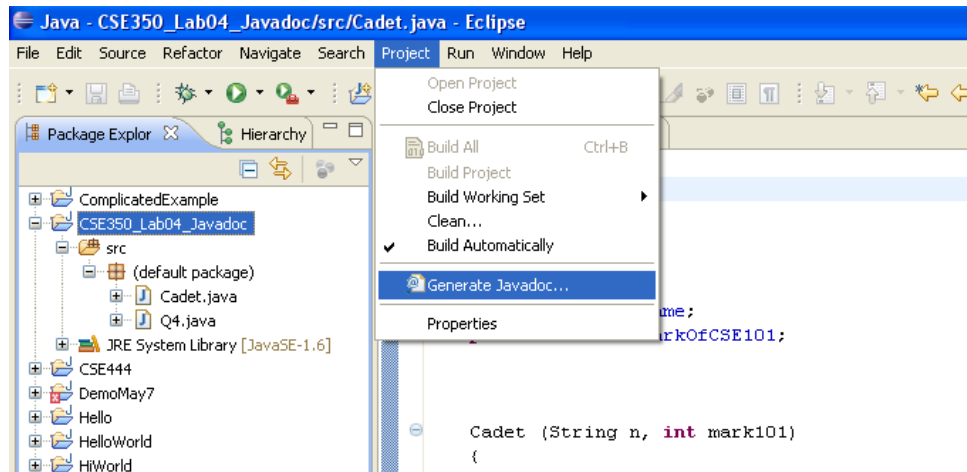
Pour plus d'informations sur les commentaires Javadoc, consultez la page Web suivante:

<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

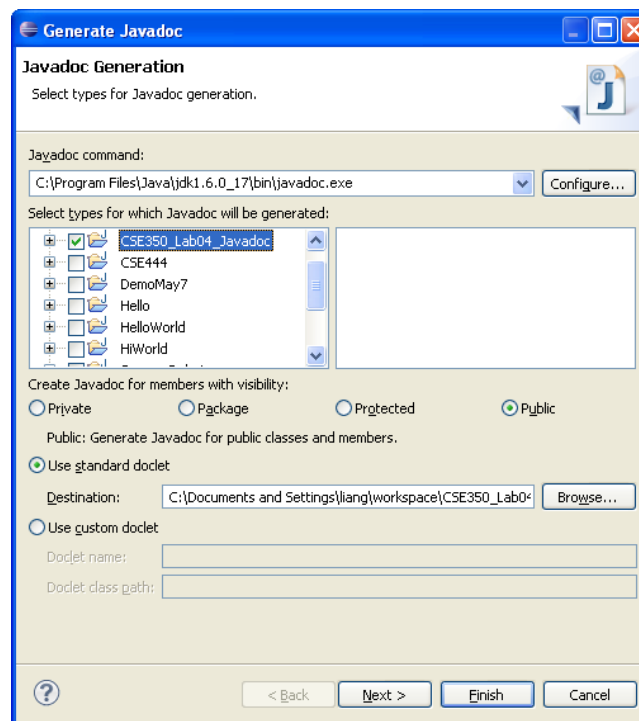
[**Tâche 2**] Lecture des éléments suivants: «Génération de Javadoc dans Eclipse IDE» et génération de Javadoc pour le code Java commenté dans la tâche 1.

Génération de Javadoc dans Eclipse IDE

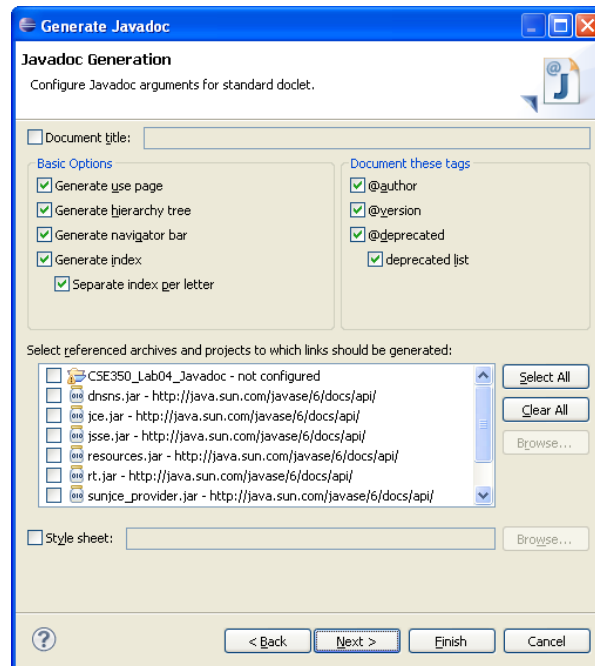
1. Créez votre projet importez les classes (fichiers Java) associées au projet.
2. Préparez des commentaires dans vos classes en suivant un style de commentaire Javadoc.
3. Cliquez sur le lien Projet et choisissez l'option «Generate Javadoc».



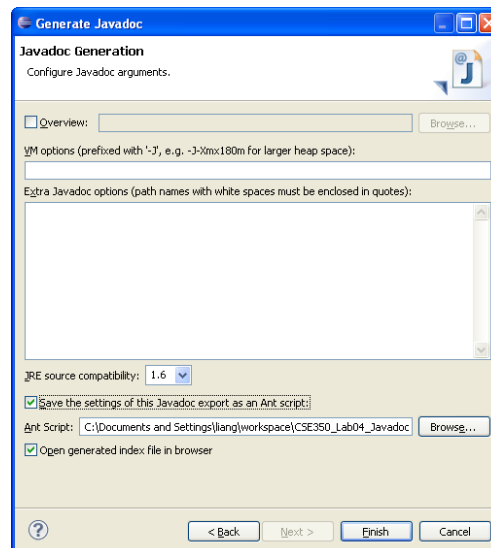
4. Maintenant, une fenêtre, «Generate Javadoc», s'ouvre pour vous permettre de sélectionner les projets Java ou leurs ressources sous-jacentes pour lesquelles Javadoc doit être généré. Plusieurs autres options sont également disponibles là où l'utilisateur peut sélectionner l'une d'entre elles selon les besoins. Ici, l'utilisateur peut choisir de générer Javadoc pour les API publiques / privées, etc.



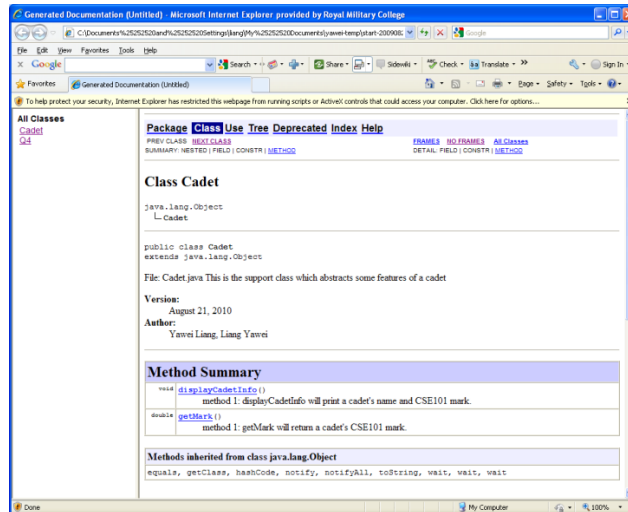
5. Désormais, IDE demande d'autres fonctionnalités pour la génération de Javadoc. Un programmeur Java doit être familiarisé avec ces options.



6. Vous pouvez également enregistrer les paramètres en tant que script ant afin que vous puissiez utiliser le script pour générer javadoc à l'avenir.



7. Cliquez sur «Terminer». Javadoc sera généré. Si vous sélectionnez l'option d'ouverture du fichier d'index dans le navigateur, après la génération de Javadoc, vous trouverez «index.htm» de Javadoc dans votre navigateur Web par défaut. Sur console, vous pouvez voir les progrès de JavaDoc Generation.



[**Tâche 3**] Revenez maintenant à votre code Java Lab02 et modifiez les fichiers Java pour inclure les commentaires appropriés, soyez prolixe. Commenter chaque classe, variable et méthode. Les commentaires doivent avoir une brève description de ce que fait le code et comment il le fait. Et utilisez les compétences acquises des tâches 1 et 2 pour générer des documentations pour ces deux programmes Java. Vous devez soumettre votre code Java commenté uniquement à partir de cette tâche.

3. Livrables et date limite

Pour le Dr Yawei Liang, vous ne devriez envoyer par courrier électronique que les fichiers Java commentés dans la tâche 3 à liang-y@rmc.ca. La date limite pour ce laboratoire est à 17h00, le 5 octobre 2018.

4. Barème

Task Description		Notes	Check (instructor)
1	Commenter correctement les classes, les variables et les méthodes.		
	Classes (auteur, version, description du contenu)	4	
	Les variables et	3	
	Méthodes	3	
Total:		10	