JGrasp and Javadoc

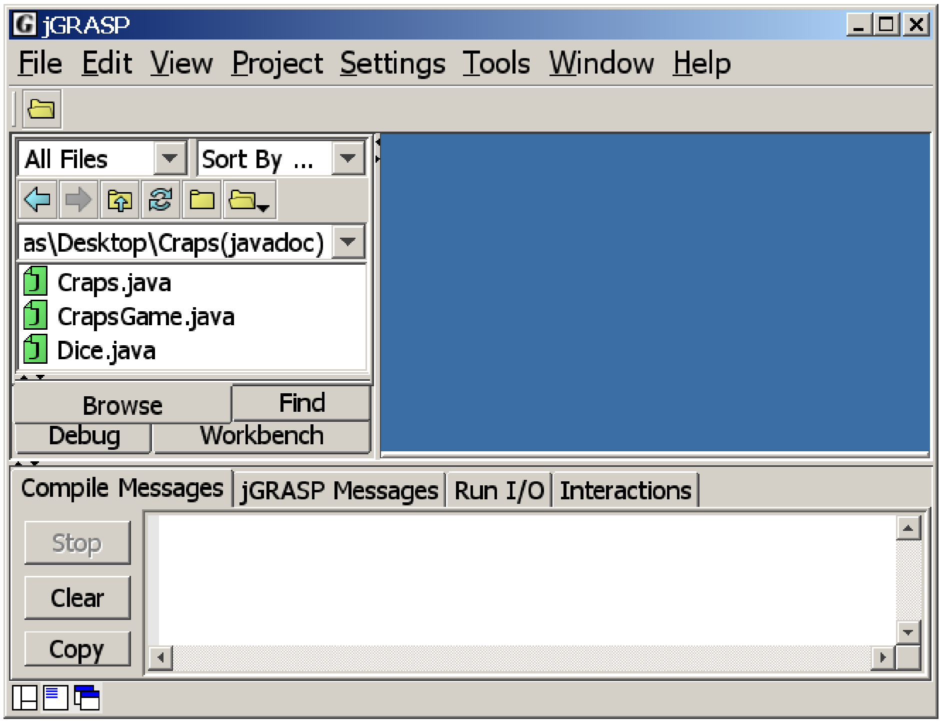
jGRASP Projects

Although you can generate javadoc without a project, since most of our programs will now be in multiple files, it makes sense to learn about the jGrasp Project. This is a programming environment feature that keeps track of all the files that belong to a particular program. Most development environments will have a feature like this. jGrasp creates a file with the extension ".gpj" , that you can conveniently double click to open all the files at once as one "project".

Creating a jGrasp Project:

Begin by taking all the craps source files (Dice.java,Craps.java, and CrapsGame.java) and put them into a folder: Craps.

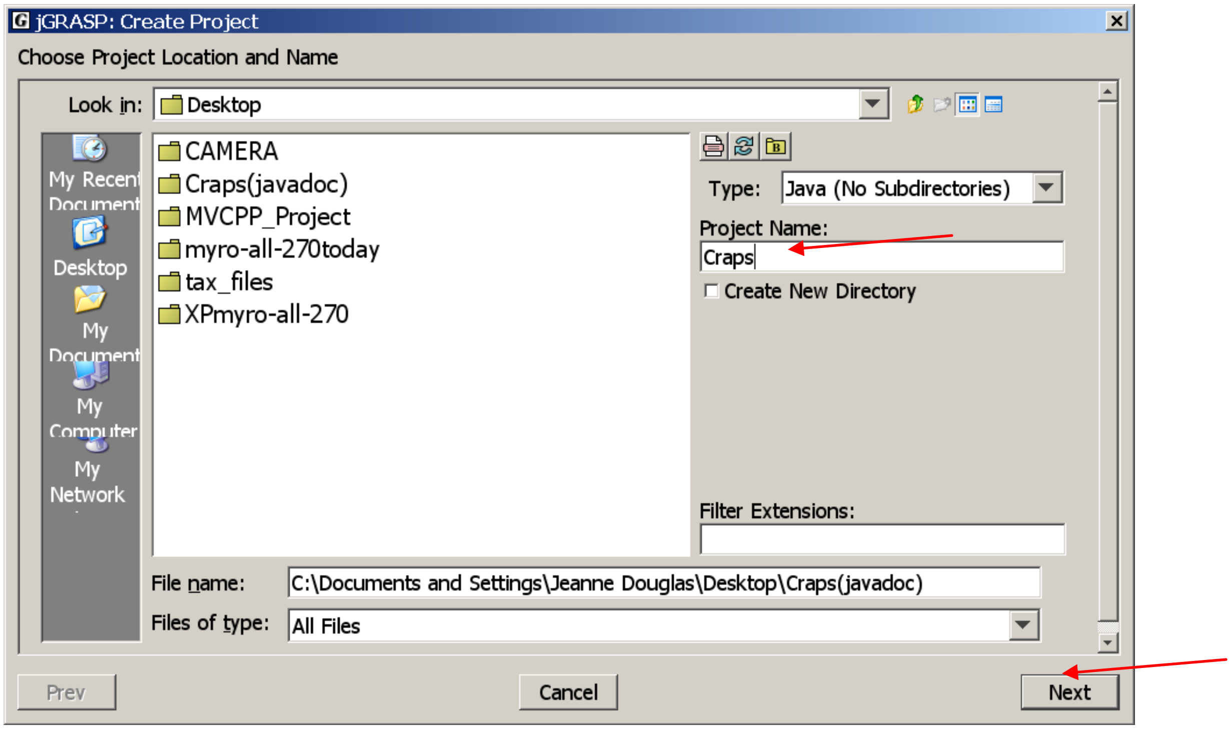
Now open JGrasp and browse to the folder Craps so you see the 3 files on the left window:



as\Desktop\Craps(javadoc)

Craps.java
CrapsGame.java
Dice.java

Create a jGrasp project:
Project/New
Type in the name of the project: Craps
Hit Next button.



jGRASP: Create Project

Choose Project Location and Name

Look in: Desktop

- CAMERA
- Craps(javadoc)
- MVCPP_Project
- myro-all-270today
- tax_files
- XPmyro-all-270

Type: Java (No Subdirectories)

Project Name: Craps

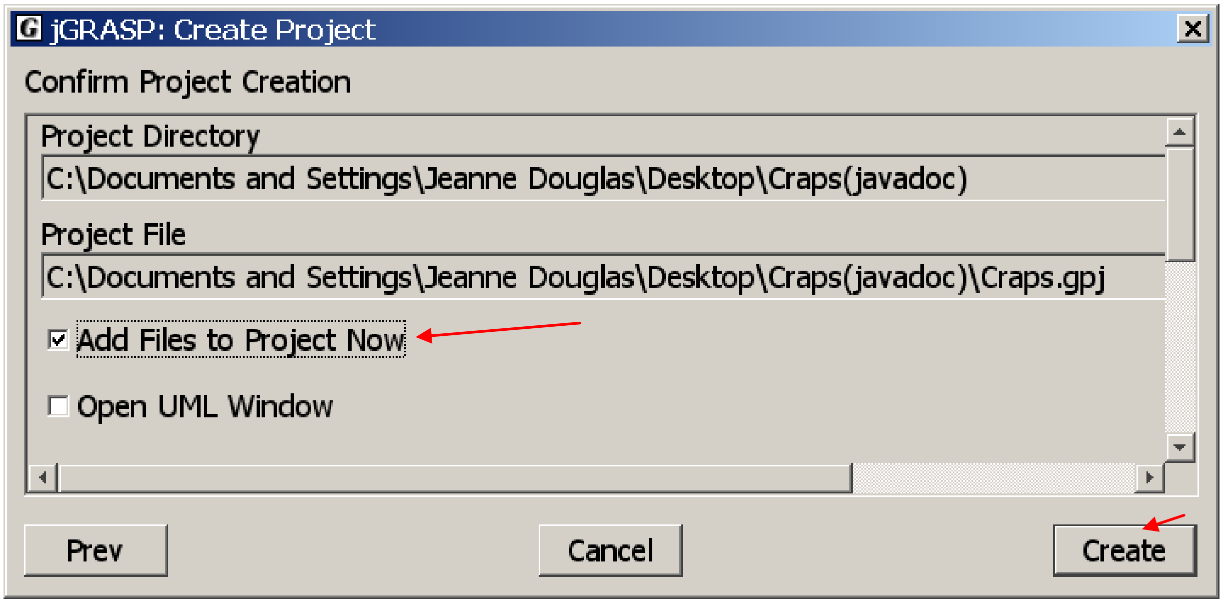
□ Create New Directory

Filter Extensions:

File name: C:\Documents and Settings\Jeanne Douglas\Desktop\Craps(javadoc)

Files of type: All Files

Select Add Files to Project Now if you already have some files ready for the project.
If no files prepared yet, then do not check this box.
Hit Create button.



jGRASP: Create Project

Confirm Project Creation

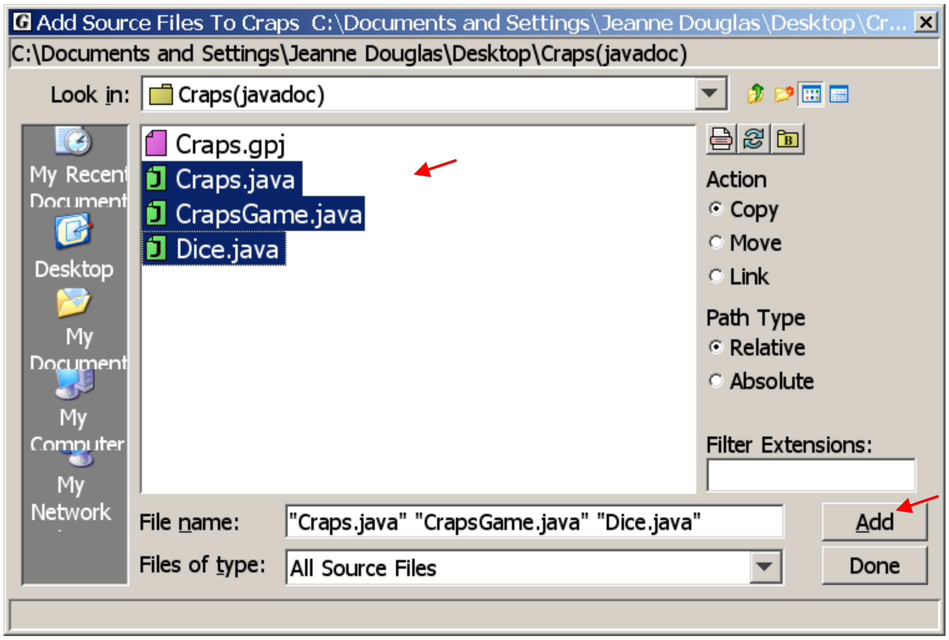
Project Directory
C:\Documents and Settings\Jeanne Douglas\Desktop\Craps(javadoc)

Project File
C:\Documents and Settings\Jeanne Douglas\Desktop\Craps(javadoc)\Craps.gpj

☑ Add Files to Project Now

□ Open UML Window

Select files to add to the project (hold CTRL key for multiple file selection).


Add Source Files To Craps C:\Documents and Settings\Jeanne Douglas\Desktop\Cr...

C:\Documents and Settings\Jeanne Douglas\Desktop\Craps(javadoc)

Look in: Craps(javadoc)

Craps.gpj
Craps.java
CrapsGame.java
Dice.java

Action: ● Copy ○ Move ○ Link

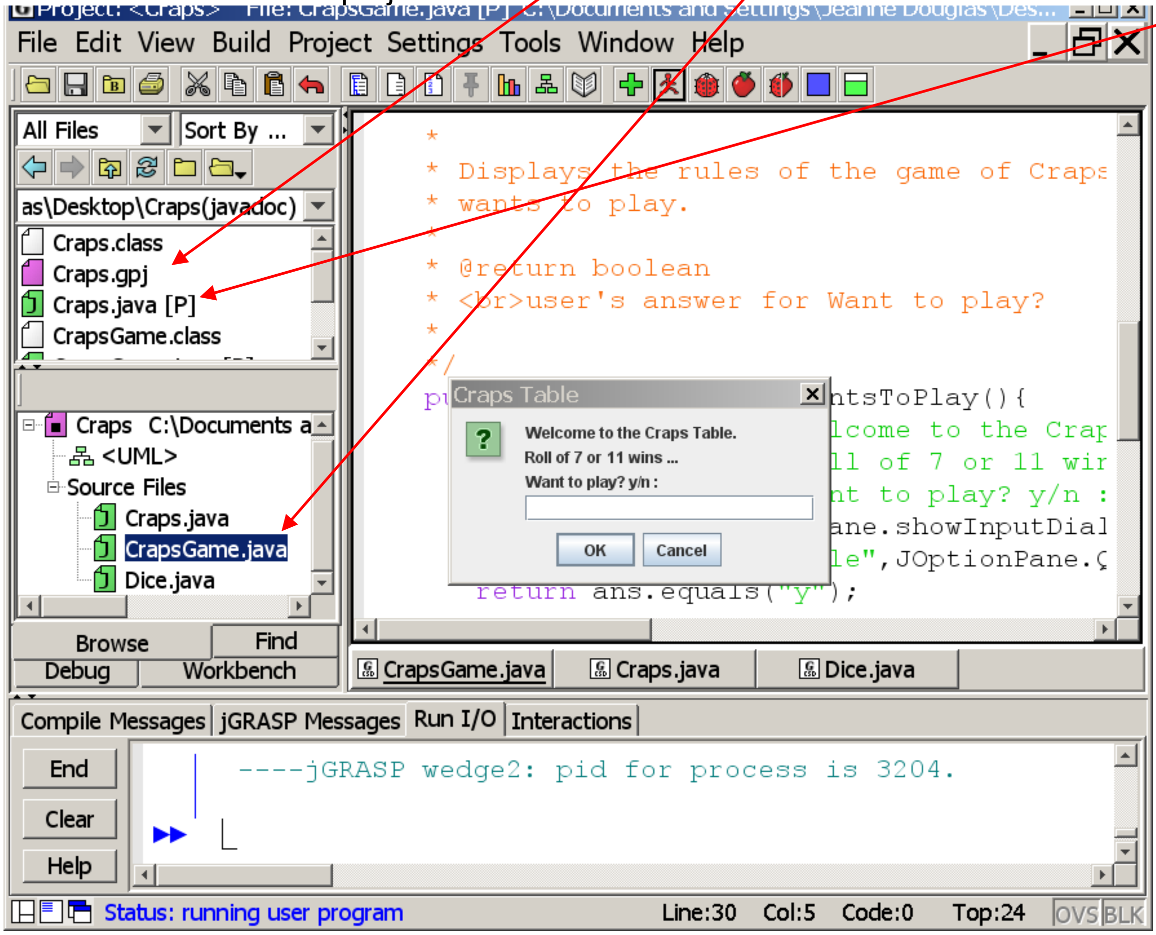
Path Type: ● Relative ○ Absolute

Filter Extensions:

File name: "Craps.java" "CrapsGame.java" "Dice.java"

Files of type: All Source Files

You should see the development window with the project open
Double click on the filenames under the project to open them.
Since more than one of our files will have “test” mains in them, the main that will run is the one that is in the file showing in the editor window. Example below shows CrapsGame running.
So, now you have a project. Notice the “Craps.jpg” file created in the folder. Also note that the [P] indicates the files in the project.


Project: <Craps>   File: CrapsGame.java [P] C:\Documents and Settings\Jeanne Douglas\Des...

```
 *
 * Displays the rules of the game of Craps
 * wants to play.
 *
 * @return boolean
 * <br>user's answer for Want to play?
 *
 */
pu                          ntsToPlay(){
                            lcome to the Crap
                            ll of 7 or 11 wir
                            nt to play? y/n :
                            ane.showInputDial
                            le",JOptionPane.Ç
    return ans.equals("y");
```

Craps Table

Welcome to the Craps Table.
Roll of 7 or 11 wins ...
Want to play? y/n :

as\Desktop\Craps(javadoc)

Craps.class
Craps.gpj
Craps.java [P]
CrapsGame.class

Craps C:\Documents a
<UML>
Source Files
Craps.java
CrapsGame.java
Dice.java

```
----jGRASP wedge2: pid for process is 3204.
```

Status: running user program   Line:30   Col:5   Code:0   Top:24

Javadoc:

Here is a link to the documentation on javadoc:
http://download.oracle.com/javase/1.4.2/docs/tooldocs/windows/javadoc.html
There are *many* features to the javadoc program. (Help yourself!)

See the practice examples (craps and bandit) where you can see what the javadoc program does with the comments. Once you know how comments are handled, you should complete all the comments consistently in your assignment so it produces clean html files of documentation.

Comments are generated differently based on the "level" of the module. For *methods*, everything up to the first "." displays in the "Method Summary". The rest of the comments are included in the "Method Detail".

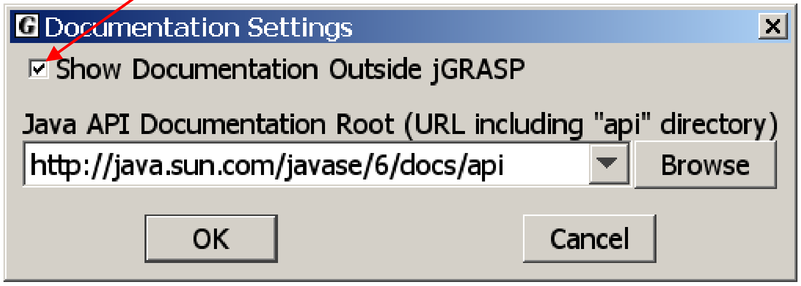
REMEMBER TO SAVE THE FILE before you regenerate the javadoc documentation, otherwise your changes in the comments won't show up! Javadoc is running on the *file* not on the editor window version which is just a “snapshot” of what is in the file.

JGrasp and Javadoc

First, some jGrasp settings. All of these may not “stick”, that is, some may need to be set up for each new project.

To show the documentation files outside of jGrasp (shows in your default browser):
Settings/Documentation Settings:
box: Show documentation Outside jGRASP
(Generally, I do NOT check this box because it’s easier to view in jGrasp than go back and forth in the browser as you’re trying to get your comments to work in javadoc .)

Also note you can pick the version of documentation you want.



Documentation Settings

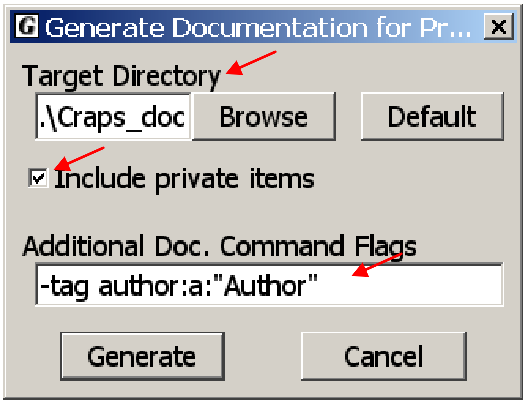
☑ Show Documentation Outside jGRASP

Java API Documentation Root (URL including "api" directory)

http://java.sun.com/javase/6/docs/api

Make the @author tag work (requires a special setting):
Project/Generate Documentation
Additional Doc. command Flags:
-tag author:a:"Author"
(This setting will have to be done for each project.)
Check box “Include private items” (you could hide these from users if you wanted).

Also, to save the files to the Target Directory, browse to a target directory.
Otherwise, jGrasp just shows you the .html files without saving them.

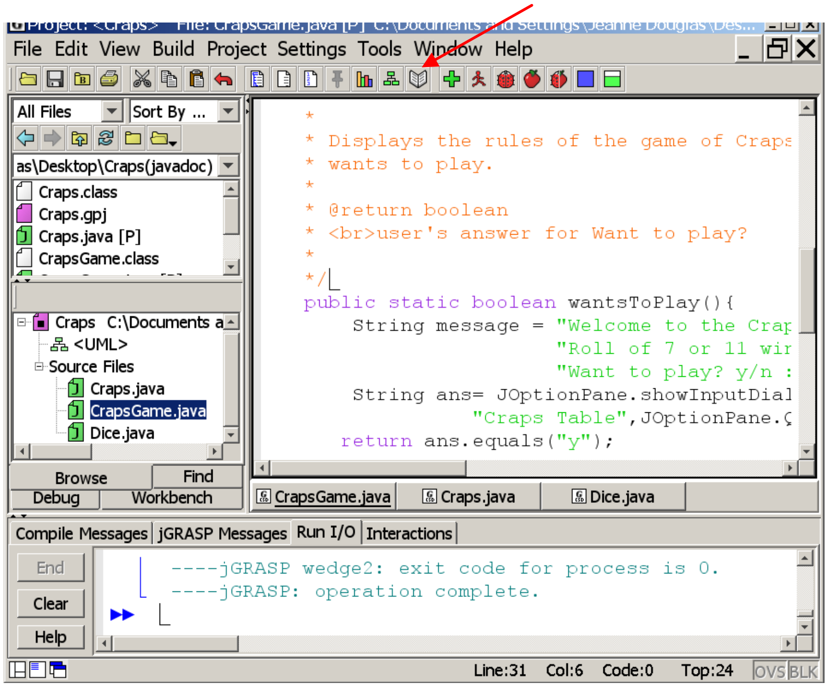


Target Directory
.\Craps_doc

☑ Include private items

Additional Doc. Command Flags
-tag author:a:"Author"

So, now you’re ready to roll.
A quick way to Generate the documentation files and show the documentation through jGrasp, hit the “book” button.



```
 *
 * Displays the rules of the game of Craps
 * wants to play.
 *
 * @return boolean
 * <br>user's answer for Want to play?
 *
 */
public static boolean wantsToPlay(){
    String message = "Welcome to the Crap
                     "Roll of 7 or 11 wir
                     "Want to play? y/n :
    String ans= JOptionPane.showInputDial
                "Craps Table",JOptionPane.Q
   return ans.equals("y");
```

```
----jGRASP wedge2: exit code for process is 0.
----jGRASP: operation complete.
```

There!
Now look over the documentation produced from the comments within the craps files.
Note *where* the comments need to appear relative to the code for javadoc to see it.
Read the comments from the code files carefully. There are hints in there to watch for.
This takes a bit of observation and trial and error work.

Expectations for our class (javadoc):

1. The class and method descriptions should be well documented to include pre/post information provided through these tags:

   tags: @author (see below)
   @return
   @param

2. You will NOT turn in the .html files generated by javadoc. The grader will simply generate them through jGrasp and look at them to see that they are well formed.

3. **.gpj files**: You do *not* need to include a jGrasp .gpj file in your assignments.
   Just make sure that you submit all the up-to-date files so your program can be compiled/executed/ and checked for javadoc.
   Best *NOT* to add files to a JGrasp project that are NOT in the same folder as this may cause you to be working on files that are part of the project but not in the correct folder for turning in.