

Forte for Java Usability Test

CVS User

User Tasks

Task 1: Explore

Start the Forte for Java Integrated Development Environment (IDE) and then spend five (5) minutes exploring. Remember to speak out loud as you explore, and try to get a general feel for how the IDE works.

Questions

1. What aspects of the IDE do you notice immediately?

2. What do you like about what you see so far?

3. What do you dislike?

Task 2: View Source Code

You are a new developer on the Java Song Player team and you need to look at the existing source code. The Java Song Player is a simple Swing user interface that allows you to play sound files. You are told that it is in the following **password protected CVS Repository**:

CVS server:	heybooboo.eng
Login name:	cvsgrp
Password:	cvsgrp
Repository path:	/usr/local/testcv
Module:	songPlayer

Your task is to **create a CVS working directory**, locate the code and briefly review it within the IDE.

Also, find out how many revisions have there been for this class.

Questions

1. How easy was it for you to create a CVS working directory?

Very Easy Easy Difficult Very Difficult

Comments on your answer:

2. How easy was it for you to locate the code?

Very Easy Easy Difficult Very Difficult

Comments on your answer:

3. Does the Forte for Java IDE give you all the information you would like about the source code?

Yes

No

If No, What other information would you like to see?

4. Enter any comments you have about this task?

Task 3: Create a New Class

1. The user interface has already been created for the Java Song Player, and you can look at it in the *SongPlayerDialog* class. Your task is to create the *PlaySong* class which provides the backend functionality for the UI to play a song by accessing a sound file from the hard drive.

Your class has the following fields:

```
private String songName; // an arbitrary name to show in dialogs
private String songFile; // the song file name
private String songPath; // the absolute path to the file
private Clip clip; // the audio clip object
```

Your class has the following methods:

```
public void play() // Start playing the song.
public void setSongInfo(final String sName, final String sPath, final String sFile) //
set up the song information and read the song from the file specified
public String getSongName() // return the name of the song
public long getSongLength() // return the length of the song
```

You'll need to import functionality from the following Java classes:

```
import java.io.*;
import javax.sound.sampled.*;
```

2. Insert the necessary code. You will find code fragments for some of these methods in the file: `c:\songcode.txt`
3. Test out the program to make sure it works as you expected.

Questions

1. Comment on your experience creating this new class:

What did you like?

What did you dislike?

2. Any additional comments related to this task?

Task 4: Share Code

Put your code into the CVS repository so that other developers can access it.

Questions

1. How easy was it to **put your code into the repository**?

Very Easy Easy Difficult Very Difficult

Comments on your answer:

2. Any additional comments on using the source control system within the IDE?

3. Are there any steps that you typically do to **put your code into the repository** that you have not done here?

Task 5: Create a Branch

A coworker of yours has been working on a progress indicator class for the song player and they're almost ready to put it in. You would like to integrate the *ProgressIndicator* class and if all goes well, then you want to put the integrated code into the source repository. Since the current code is working well, you want to create a branch with the tag name "update" so that you can update the version of the Java Song Player program to include the progress indicator without affecting the working code in the main branch of the CVS repository.

Questions

1. Would you use this functionality from within the IDE?

Yes

No

If No, why not?

2. Any additional comments on this task?

Task 6: Get Updates

Your coworker has put their code into **the CVS repository**. Your task is to update your **working directory** with the new code from the **CVS repository**. Briefly review the code.

Question

1. How easy was it to get your coworker's code?

Very Easy Easy Difficult Very Difficult

Comments on your answer:

Task 7: Integrate Code

To integrate the new *ProgressIndicator* class into the Java Song Player, you need to do the following:

1. Add a new *ProgressIndicator* object field in the *SongPlayerDialog* class.
`private ProgressIndicator progInd;`
2. Add a line at the end of the constructor method for the *SongPlayerDialog* class to create a *ProgressIndicator* object:
`// create a progress indicator object to show while the song is playing
progInd = new ProgressIndicator((java.awt.Dialog)this, true);`
3. Show the progress indicator when the song starts to play by adding the following code to the end of the **playButtonActionPerformed** method in the *SongPlayerDialog* class:
`// show the progress indicator while the song is playing
progInd.showProgress(mySong.getSongName(), mySong.getSongLength());`

Modify the existing classes as above and then test to make sure it all works correctly.

Questions

1. Any comments on your experience performing this task?

Task 8: Create a JAR File

Combine the code in the package into a single Java Archive (JAR) file so that you can easily send it to others outside your team for review.

Question

1. How easy was it to create the JAR file?

Very Easy Easy Difficult Very Difficult

Comments on your answer:

Task 9: Merge Branch

Now you need to commit the integrated code and then merge the “update” branch with the main branch of the CVS source code repository in order to complete your work on the Java Song Player.

This requires the user to first do a CVS -> Commit on the songPlayerDialog class, then do a CVS -> Merge with Branch

Some Final Questions

1. Do you typically develop using an IDE such as Forte for Java?

- Yes
- No

2. If yes, which of the following tasks do you feel that you are most productive using an IDE? (check all that apply)

- Creating user interface code (GUI)
- Editing code (please list types/languages): _____
- Compiling code
- Building programs
- Debugging
- Others (please list): _____

3. Briefly describe your use of source control. Discuss the point at which code is placed under source control, how often you refresh your code from the repository, and how often you write code that doesn't go into the repository.

4. If you accessed the online help during this session, please comment on its helpfulness. If you haven't accessed the online help, please take a look at the help menu. Would you be likely to access this online help?

- Yes No Why or why not?

5. Would you recommend the use of Forte for Java IDE to coworkers or friends doing Java development?

- Yes
- No
- Why?

6. Would you recommend the use of **CVS** source control within the Forte for Java IDE to coworkers or friends doing Java development?

- Yes
- No
- Why?

7. If you had only a few minutes to explain this IDE and **CVS** source control tool to someone, what would you say?