**Escuela de Ingeniería Informática de Oviedo** **Second theory exam.** **December-2015**

**Human Computer Interaction model 1**

*TEST: Right answer +4 , wrong answer -1, blank +0.* ***There is one only right answer per question.***

|  |
| --- |
| **Name and Surname:** |
| **ID**: |

**package geometria;**

…

class Aplicacion{

public static void main(String[] args){

…

Circulo c1 = new Circulo();

c1.calcularArea();

…

}

}

1. . **Giving the code above, and considering that:**

* The code corresponding the class Circulo is included in the file *figuras.jar* and located in the folder *C:\utilidades*
* The application *(Aplicacion.class)* is located in the folder *C:\Archivos de programa\aplicaciones\ geometria*

Which is the right value for the classpath?

* 1. Classpath=C:\Archivos de programa\aplicaciones\geometria; C:\utilidades\figuras.jar
  2. Classpath=C:\Archivos de programa\aplicaciones; C:\utilidades\figuras.jar
  3. Classpath=C:\Archivos de programa\aplicaciones; C:\utilidades
  4. Classpath=C:\Archivos de programa\aplicaciones\geometria; C:\utilidades
  5. None of the answers is right.

1. **We have a JavaHelp based help system that has an index and search views (there is no table of contents), and we must extend it adding a new html file. What must we do in order to reflect the new file in both views and to use it as the default page of the help system?**
   1. Modify Ayuda.hs and execute jhindexer
   2. Modify the toc file, the index file and execute hsviewer
   3. Modify the map file, the index file, the Ayuda.hs file and execute jhindexer.
   4. Modify the map file, the toc file and the index file.
   5. The table of contents is provided by default and cannot be removed in JavaHelp
2. **The *indispensable* files we need to build a JavaHelp based help system are:**
   1. Every html and xml files
   2. Every html and xml files and the help index database.
   3. Every file except the helpset file and the help index database.
   4. Every html file, the map file and the helpset file.
   5. None of the files is indispensable.
3. **The method we must use in java in order to enable the context sensible help on a component is:**
   1. enableHelp(…)
   2. enableHelpKey(…)
   3. enableHelpOnButton(…)
   4. Redefine the method *keyPressed* of the keyboard event on the component sending the key *VK\_F1* as an argument
   5. None of them is right.
4. **Regarding the creation of automatic installers, which of the following sentences is true?**
   1. The .project file of the project must be included in the installer.
   2. Whenever the application uses JavaHelp, the corresponding jar file with the classes we need (*jhall.jar during the lab)* file must be included in the installer.
   3. We cannot set up the classpath of the application from the installation process, this must be configured manually after the installation.
   4. None of the answers is right.
   5. More than one answer is right
5. **The helpset file (ayuda.hs) of JavaHelp…**
   1. Associates ids or alias to the html files.
   2. It is the file explicitly referenced from the application.
   3. It describes the content and distribution of the table of contents.
   4. It states the views (table of contents, index, and searcher) that will be provided to users in the help window.
   5. More than one answer is right.
6. **After the “deep” study we made during the seminar regarding the internationalization of different web sites, we can conclude that…**
   1. Most of the analyzed sites are perfectly and completely internationalized.
   2. All the Town halls were perfectly internationalized and prepared for at least three different localizations each.
   3. Probably due to the economic impact of the web in their business, the Hotels seem to have better internationalized web pages than the other two groups, even when some of them have still very serious issues.
   4. None of the answers is right.
   5. More than one answer is right.
7. **In a menu…**
   1. Radio buttons cannot be used. Exclusive options must be shown by means a submenu that shows a new dialog.
   2. If every item of the menu is disabled, the menu title must be also disabled.
   3. **Every** option in the menu should have accelerator key or shortcut.
   4. More than one answer are right
   5. None of the answers is right.
8. **Regarding layouts…**
   1. The default layout of a JPanel is *GridLayout*.
   2. A panel that uses a *GridBagLayout* is divided in a number of cells with identical size.
   3. JToolBar is an example of *BoxLayout*
   4. If a container uses *BorderLayout*, the components can modify their position by means of the *position* property.
   5. If we want to divide a panel in a number if cells with different sizes, the most adequate layout is *FlowLayout.*
9. **Regarding the internationalization of Java applications, we can state that…**
   1. Texts, numbers, currencies and dates to be localized are included in the resource bundles (properties files).
   2. Texts are stored in the resource bundles (properties files), while numbers, currencies and dates are formatted by means the corresponding sentences in the source code.
   3. Texts, numbers, currencies and dates are formatted and “localized” by means the corresponding sentences in the source code, and **optionally** we can also use resource bundles to optimize the localization process.
   4. To adapt the images to the different localizations we need to store every image of the same localization inside a file named as *baseName\_localization* (example: images\_en, images\_es).
   5. None of the answers is right.
10. **If we have a correctly and completely internationalized application in Java, and we want to extend it with a new localization…**
11. We don’t have to do anything, Google translate will do that automatically for us
12. We will be forced to change the source code (because of the currency, date and number formats) and add new property files to the project.
13. We will have to do some additions to the project, but without recompiling the java classes at all
14. It is not possible to add new localizations
15. None of the answers is right.
16. **If we want to include the search view in our JavaHelp based help system…. Which of the following steps are necessary?**
17. We modify the jhindexer.xml file adding the words we must include in the search
18. We execute hsviewer over the folder containing the help html files
19. We execute jhsearch over the folder containing the help html files
20. We execute jhindexer over the folder containing the help html files
21. We add the search view in the helpset file, specifying the folder that contains the help html files.
22. **Regarding the JList component, we can state that….**
    1. It allows users to select one single element of type String (given that the toString method is invoked during JList rendering)
    2. It allows users to select one single object
    3. It allows users to select one or more objects of type String (given that the toString method is invoked during JList rendering)
    4. It allows users to select one or more objects
    5. We must use it always in combination with a JFileChooser.
23. import java.util.\*;
24. class Ejemplo{
25. public static void main ( String[] args){
26. Locale localizacion = Locale.getDefault(Locale.Category.FORMAT);
27. ResourceBundle mensajes = ResourceBundle.getBundle("Textos",localizacion);
28. System.out.println(mensajes.getString("t1"));
29. System.out.println(mensajes.getString("t2"));
30. }
31. }
32. **. Considering the code shown above is internationalized for French and English, and that we need to internationalize it also for Spanish…. We should:**
33. Modify Textos\_fr\_en.properties adding the new texts in Spanish and changing the name to Textos\_fr\_en\_es.properties.
34. Create a new file named Textos\_es.properties.
35. Add a new line of code: System.out.println(mensajes.getString("t3"));
36. Modify Textos\_fr\_en.locale adding the new texts in Spanish and changing the name to Textos\_fr\_en\_es.locale.
37. Create a new file named Textos\_es.properties and add a new line of code: System.out.println(mensajes.getString("t3"));
38. **In JavaHelp, the map file:**
39. Defines the available views in the help system.
40. It describes the content and distribution of the table of contents.
41. Associates identifiers or aliases to HTML files.
42. It describes the content and distribution of the index.
43. More than one answer are right