**The Experimentation Scheme of Jukebox**

卓一 2016302580095 余连玮

Create Song class and Jukebox class and run them. General scheme: Jukebox should accept input from *JukeboxForm.html* and create an HTML page that redisplays the song list as an HTML form, and adds a text area that displays the user's selections. To create Jukebox, the following steps can be taken:

1. Create Song class. The Song class models a song with the following attributes: title, play time, category, album/track, ID (the same as *the value of checkbox* in the *JukeboxForm.html* ), link to audio file, and whether the song has been selected by the user. The class should have public methods to set the values of attributes. The class should also have public methods (that is *get methods*) to return the values of attributes.
2. Finish the doPost method:
   * Before write the doPost method, create three Song objects with attributes of the songs on *JukeboxForm.html*.
   * Create a Vector named songlist and add the three *new* Song objects(*using the keyword “new” to create three Song objects*) to it by *using Vector’s method songlist.add()*
   * Get user input (*declare a string variable named “titleChoice” to store the input data*) by using HttpServletRequest’s method getParameter(). Since the value of the input control is the songs’ ID, we should use Vector’s *elementAt(int index) method and getSongID() method* to get the songs’ ID one after another(*use the For-Loop to do it and declare int “lcount” for loop counter*).
   * Update the collection of Song's for selection status. Use *the control flow If- Statement (if the titlechoice has value)* to uplate the selected songs’ status. For each song in the Vector, change the value of the instance variable that indicates whether the song has been selected by the user.(*Let Boolean selected be True*).
   * Prepare the HTML and send it out by *using displayHtml (songList, response) method.*
3. Finish the displayHtml method:
   * Indicate the content type (text/html) being returned by the response.*Use HttpServletPesponse’s setContentType() method.*
   * Retrieve an output stream (*a PrintWriter object named out1*) to use to send data to the client by *using getWriter() method*.
   * Build the HTML page. Create an instance of the HtmlPage class.
   * Set the attributes of the HtmlPageV2 object. Just set the same atttibutes as the initial JukeboxForm.html by *using the setTitle, setBackgroundColor and addText methods.*
   * Continue to build the HTML page by *adding the list of songs using the HtmlTable class*. *Use the startRow, addCell, and endRow methods of HtmlTable class to build the table*.
   * Then use a loop to create the checkbox for the song titles to choose from and populate the body of the table with the properties of the Song objects in the Vector.
   * When adding a cell that contains a link, use instances of the HtmlAnchor class to generate the HTML for the link.
   * Then using *HtmlTable’s bulidHtml()* method to complete the table.
   * Add the text that creates the Submit button. (*Use the addText() method.*)
   * Display the playlist in a **TEXTAREA**. You will need a loop to display the play list. *Use the control flow If- Statement and Song’s isSelected() method* to dieplay the songs that are selected.
   * Add the closing **TEXTAREA** tag and FORM tag using the addText method.
   * Construct the HTML response. *Use print, flush and close method.*
   * *Finish!*
4. Run the servlet on the Tomcat server.