**Additional Notes**

The first class was the PlayListControlsPanel. We noticed that this class implemented the JPanel class.

This is where the "sortBy" items were defined. Each one had a separate method that called its own method from the controller. So this led us to think that for our new menu should follow this pattern of the panel in the view being connected to the controller.

The PlayListController had a lot of methods involved in it, not to mention the plethora of Imported libraries. This was semi-familar concept because of our experience with other languages. We had to walk through the code for each method before identifying the area where the Column clicks were taking place. We marked that location and traced to the next class.

The PlayListControlsListener had the proxy calls to do the various sorts. We wanted to keep things consistent, so we marked that location and traced to the next class.

Several remarks about the similarity between Java and C#. Method names, similar construction of code, generics, etc.

The PlayListHandler class is where we first encountered the calls to the Comparator pattern used by Atunes.

Since the Duration comparator did not exist for the feature we were tasked with implementing, we copied out one of the other comparator classes, and then used it to create our PlayListDurationCompartor.java class.

The AudioFile was an interesting class during this step. We clicked on definition and looked up the methods that it had. We were pleased to discover the **getDuration** method. That made our comparisons very easy to implement.

**Class references**

|  |
| --- |
| public class **PlayListController** extends PanelController implements PlayListColumnClickedListener  public PlayListController(PlayListPanel panel)  public void finish()  protected void addBindings()  protected void addStateBindings()  protected void notifyReload() {}  public void columnClicked(String column)  public void updatePositionInTable(int pos)  public void addSongsToPlayList(ArrayList songs, int selectedSong)  public void setSelectedSong(int song)  private void setSelectedSongOnTable(int song)  public void adjustColumnsWidth()  public void playSelectedSong()  public void setTrackNumber()  public void setGenre()  public void moveToTop()  public void moveUp()  public void moveDown()  public void moveToBottom()  public void deleteSelection()  public void setAsFavoriteSongs(ArrayList<AudioFile> songs)  public void setAsFavoriteAlbums(ArrayList<AudioFile> songs)  public void setAsFavoriteArtists(ArrayList<AudioFile> songs)  public void disablePlayListItems(boolean disable)  private ArrayList<AudioFile> getSelectedSongs(JTable table)  public ArrayList<AudioFile> getSelectedSongs()  public JScrollPane getMainPlayListScrollPane()  public PlayListTable getMainPlayListTable()  public void setArtistAsPlaylist()  public void setAlbumAsPlaylist()  public void showPlaylistControls(boolean show) |

|  |
| --- |
| public class **PlayListListener** extends MouseAdapter implements ActionListener, ListSelectionListener  private PlayListTable table;  private PlayListController controller;  protected PlayListListener(PlayListTable table, PlayListController controller)  public void actionPerformed(ActionEvent e)  public void mouseClicked(MouseEvent e)  public void valueChanged(ListSelectionEvent e) |

|  |
| --- |
| public class **PlayListHandler**  static Logger logger = Logger.getLogger(PlayListHandler.class);  private static final String M3U\_HEADER = "#EXTM3U";  private static final String M3U\_START\_COMMENT = "#";  private PlayList nonFilteredPlayList;  private final PlayListListener playListListener = new PlayListListener()  public PlayListHandler()  public void editTags()  public void savePlaylist()  public void loadPlaylist()  public void getLastPlayList()  public Runnable getLoadPlayListProcess(final ArrayList<String> files)  public static final FileFilter getPlaylistFileFilter()  private void sortPlayList(Comparator comp)  public void sortPlaylistByTrack()  public void sortPlaylistByTitle()  public void sortPlaylistByArtist()  public void sortPlaylistByAlbum()  public void sortPlaylistByGenre()  public void sortPlaylistByDuration()  private static ArrayList<String> read(File file)  private static boolean write(PlayList playlist, String fileName)  public static ArrayList<AudioFile> getFilesFromList(File file)  public PlayListListener getPlayListListener()  public void setFilter(String filter)  private void setPlayListAfterFiltering(PlayList playList)  public void moveToTop(int[] rows)  public void moveUp(int[] rows)  public void moveDown(int[] rows)  public void moveToBottom(int[] rows)  public void removeSongs()  public void removeSongs(int[] rows)  public void addToPlayList(ArrayList<AudioFile> files)  public boolean isFiltered()  private void setPlayList(ArrayList<AudioFile> files)  public void finish()  public void clearList() |

|  |
| --- |
| public class **PlayListDurationComparator** implements Comparator  public static PlayListDurationComparator comparator = new PlayListDurationComparator();  public int compare(Object o1, Object o2) {  AudioFile f1 = (AudioFile) o1;  AudioFile f2 = (AudioFile) o2;  Long l1 = f1.getDuration();  Long l2 = f2.getDuration();  return (int) l1.compareTo(l2);  } |

|  |
| --- |
| public class **AudioFile** extends File  private static Logger logger = Logger.getLogger(AudioFile.class);  private static final long serialVersionUID = 7907490762633634241L;  public static final String MP3\_FORMAT = "mp3";  public static final String OGG\_FORMAT = "ogg";  public static final String MP4\_FORMAT\_1 = "m4a";  public static final String MP4\_FORMAT\_2 = "mp4";  public static final String MP4\_FORMAT\_3 = "aac";  public static final String WAV\_FORMAT = "wav";  public static final String WMA\_FORMAT = "wma";  private Tag tag;  private ArrayList externalPictures;  private long duration;  private int bitrate;  private int frequency;  public AudioFile(String fileName)  private void introspectTags()  public int getExternalPicturesCount()  public void setExternalPictures(ArrayList externalPictures)  public void addExternalPicture(File picture)  public void refreshTag()  public String toString()  private void deleteTags()  public String getNameWithoutExtension()  public String getTitleOrFileName()  public String getTitle()  public String getArtist()  public String getAlbum()  public Integer getTrackNumber()  public String getGenre()  public final boolean hasInternalPicture()  public void setTag(Tag tag)  public Tag getTag()  public static boolean isValidAudioFile(File file)  public static boolean isMp3File(File file)  public static boolean isOggFile(File file)  public static boolean isMp4File(File file)  public static boolean isWavFile(File file)  public static boolean isWmaFile(File file)  public static Tag getNewTag(AudioFile file, HashMap properties)  public ArrayList getExternalPictures()  private static void readAudioProperties(AudioFile file)  public long getDuration()  public int getBitrate()  public int getFrequency() |