

Practical4

Name: Xiaosheng Liang

Date: October 21, 2016

1) In **owner.cvs**, I have changed some songs to be my own and make sure the song list corresponds to what I have done.

	A	B	C	D	E	F
1	Songid, "Libraries"					
2	# THIS FILE CONTAINS Mapping from Song-Ids to Owners					
3	# Aug 30 2014					
4	#####					
5	# All SONGS					
6	#####					
7	10, "apple markk xiaosheng"					
8	11, "apple markk xiaosheng"					
9	12, "apple markk xiaosheng"					
10	13, "apple markk"					
11	14, "apple markk"					
12	15, "apple markk"					
13	16, "apple xiaosheng"					
14	17, "apple markk"					
15	18, "apple markk xiaosheng"					
16	19, "apple"					
17	20, "apple"					
18	21, "apple"					
19	22, "apple"					
20	23, "apple"					
21	24, "apple"					
22						
23						

2) For **build_all** method in **album.rb**, I have defined the method like this.

```

# Class Method that builds albums from song object's contents.
# It returns an array of album objects. It calls another class method that
# builds a single album, given the name of that album.
def self.build_all(albums = [])
  for i in 0...$songs.length-1
    albums[i]=$songs[i].album          #search all the albums in .csv file
  end
  albums=albums.clean_up
  for i in 0...albums.length-1
    build_an_album_called(albums[i])    #create the album array without repeated album
  end
  # p albums
  return albums
end

# Class method that takes an album name, it finds all the sounds that are in that album
# builds up arrays of the song-names (tracks), runtimes, artist names. These all get used
# to populate the various attributes of the album object.
def self.build_an_album_called(album_name,song_detail=[])
  for i in 0...$songs.length-1
    if $songs[i].album==album_name
      # push the song_name,song_id,runtimes,artists into each album array
      song_detail.push($songs[i].name,$songs[i].id,$songs[i].time,$songs[i].artist)
    end
  end
  # p song_detail
  return song_detail
end
end

```

I got all the information from `$songs[]` and pick the name of albums into a new array, and deleted the repeated, that's **albums**. And calling the method **build_an_album_called** to build a single album and add details (include song_name, id, runtimes, artist).

3) By using the method **clean_up()**, the names of actor could be get from **\$hash_owners** and remove repeated ones.

```

class Array
  def clean_up()
    new_array=[self[0]]          #create a new array
    $i=1
    $num=self.length
    while $i<$num
      if self[$i]!=new_array[new_array.length-1]
        new_array.push(self[$i])  #push new album name into new array
      end
      $i+=1
    end
    return new_array
  end
end
end

```

4) In **reader.rb** , when we read the song_id from 2 .csv files, we should check whether it is a number, if it's a number, then check whether it's in the range of 10 to 24. Finally, adding the song_id in **\$songs** and **\$hash_owners**.

```
# method to read in owners and the ids of the songs they own from the
# owners.csv file.
# returns a hash table where the keys are song-ids and the values are the owners
# this song (a string)

def read_in_ownership(csv_file_name, temp_hash = Hash.new)
  CSV.foreach(csv_file_name, :headers => true) do |row|
    song_id, owner_data = row[0], row[1]
    unless (song_id =~ /\d/)
      song_id=song_id.to_i      #transform the type of song_id to Fixnum
      if (song_id != "Fixnum")
        puts "Error: the type of id is wrong."
      elsif(song_id>24 && song_id<10)
        puts "Error: the number isn't the id."
      end
      temp_hash[song_id] = owner_data
    end
  end
  temp_hash
end
```