Category Enum

```
public enum Category {
	COMEDY, EDUCATION, ENTERTAINMENT, SPORTS, OTHER
}
```

Class Video

```
public class Video {

	private int id;
	private String title;
	private int duration;
	private Category category;

	public Video(int id, String title, int duration, Category
category) {
		this.id = id;
		this.title = title;
		this.duration = duration;
		this.category = category;
	}

	public int getDuration(){
		return duration;
	}
	public Category getCategory() {
		return category;
	}
	public boolean equals(Object obj){
		if(this == obj)
			return true;
		if(obj == null)
			return false;
		Video temp;
		if(obj instanceof Video)
			temp = (Video) obj;
		else
			return false;
		if(this.id == temp.id)
			return true;
		return false;
	}
	@Override
	public String toString() {
		return "Video [id=" + id + ", title=" + title
	+ ", duration=" + duration + ", category=" + category + "]";
	}
}
```

Class Playlist

```
import java.util.Arrays;

public class Playlist {

	private String name;
	private boolean shared;
	private Video videos[];
	private int nbVid;

	public Playlist(){
		videos = new Video[0];
	}

	public Playlist(Playlist p){ //Copy constructor
		this.name = p.name;
		this.shared = p.shared;
		this.videos = new Video[p.videos.length];
		for(int i = 0; i < p.nbVid; i++)
			this.videos[i] = p.videos[i];
		this.nbVid = p.nbVid;
	}

	public Playlist(String name, boolean shared, int size){
		this.name = name;
		this.shared = shared;
		videos = new Video[size];
		nbVid = 0;
	}
	public int getIndexOf(Video v){
		for(int i = 0; i < nbVid; i++)
			if(videos[i].equals(v))
				return i;
		return -1;
	}
	public boolean addVideo(Video v){
		int index = getIndexOf(v);
		if(index != -1 || nbVid == videos.length)
			return false;
		videos[nbVid++] = v;
		return true;
	}
```

```java
public boolean removeVideo(Video v){
	int index = getIndexOf(v);
	if(index == -1) return false;

	for(int i = index; i < nbVid-1; i++)
		videos[i] = videos[i+1];

	videos[nbVid-1] = null;
	nbVid--;
	return true;
}

public int countVideosOf(Category cat){
	int counter = 0;
	for(int i = 0; i < nbVid; i++)
		if(videos[i].getCategory() == cat)
			counter++;
	return counter;
}
public void sortOnDuration(){  //Bubble Sort
	for(int i = 0; i < nbVid - 1; i++){
		for(int j = 0; j < nbVid - 1 - i; j++){
			if(videos[j].getDuration() >
				videos[j+1].getDuration()){
				Video temp = videos[j];
				videos[j] = videos[j+1];
				videos[j+1] = temp;
			}
		}
	}
}
public String getName() {
	return name;
}
public boolean isShared() {
	return shared;
}
public void setShared(boolean shared) {
	this.shared = shared;
}
public boolean equals(Playlist p){
	return this.name.equalsIgnoreCase(p.name);
}
@Override
```

```
	public String toString() {
		return "Playlist [name=" + name + ", shared=" + shared
		+ ", videos=" + Arrays.toString(videos) + "]";
	}
}
```

Class Channel

```
import java.util.Arrays;

public class Channel {

	private String name;
	private Playlist playlists[];
	private int nPlay;

	public Channel(String name, int size){
		this.name = name;
		playlists = new Playlist[size];
		nPlay = 0;
	}
	public int getIndex(Playlist p){
		for(int i = 0; i < nPlay; i++)
			if(playlists[i].equals(p))
				return i;
		return -1;
	}
	public boolean addPlaylist(Playlist p){
		int index = getIndex(p);
		if(index != -1 || nPlay >= playlists.length)
			return false;
		playlists[nPlay++] = new Playlist(p);
		return true;
	}
	public boolean deletePlaylist(Playlist p ){
		int index = getIndex(p);
		if(index == -1)
			return false;
		playlists[index]  = playlists[nPlay-1];
		playlists[nPlay-1] = null;
		nPlay--;
		return true;
	}
	public boolean flipShared(Playlist p){
		int index = getIndex(p);
		if(index == -1)
			return false;
		playlists[index].setShared(!playlists[index].isShared());
		return true;
	}
```

```
	public Playlist getPlaylist(Category cat){
		if(nPlay == 0)
			return null;
		Playlist max = playlists[0];
		for(int i = 1; i < nPlay; i++)
			if(playlists[i].countVideosOf(cat) >
					max.countVideosOf(cat))
				max = playlists[i];

		return max;
	}

	public void sortPlaylists(){
		for(int i = 0; i < nPlay; i++)
			playlists[i].sortOnDuration();
	}

	@Override
	public String toString() {
		return "Channel [name=" + name + ", playlists=" +
Arrays.toString(playlists) + ", nPlay=" + nPlay + "]";
	}
}
```