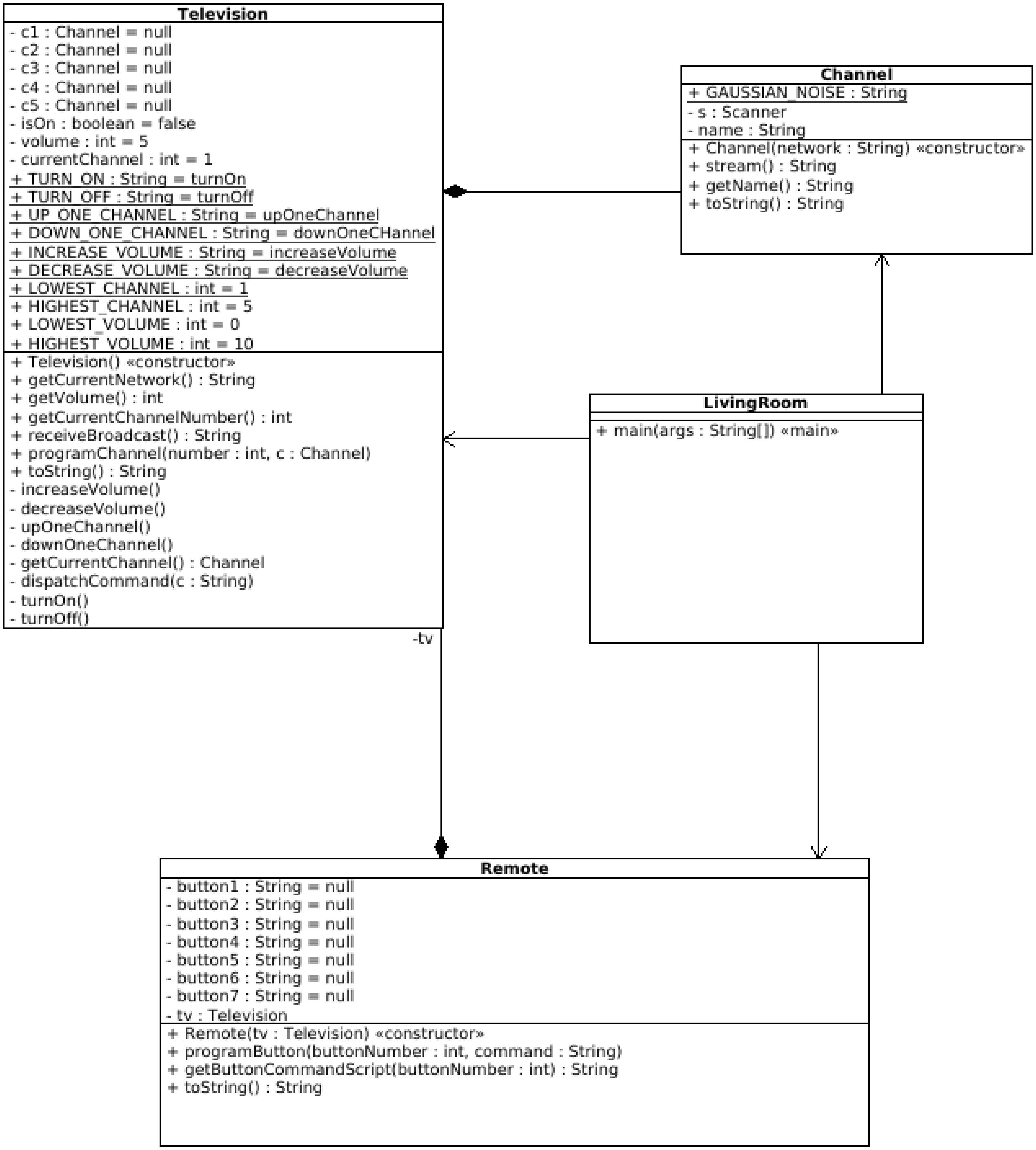
CSC110 Fall 2019 Final Code

UML Diagram


Television
- c1 : Channel = null
- c2 : Channel = null
- c3 : Channel = null
- c4 : Channel = null
- c5 : Channel = null
- isOn : boolean = false
- volume : int = 5
- currentChannel : int = 1
+ TURN_ON : String = turnOn
+ TURN_OFF : String = turnOff
+ UP_ONE_CHANNEL : String = upOneChannel
+ DOWN_ONE_CHANNEL : String = downOneCHannel
+ INCREASE_VOLUME : String = increaseVolume
+ DECREASE_VOLUME : String = decreaseVolume
+ LOWEST_CHANNEL : int = 1
+ HIGHEST_CHANNEL : int = 5
+ LOWEST_VOLUME : int = 0
+ HIGHEST_VOLUME : int = 10
+ Television() «constructor»
+ getCurrentNetwork() : String
+ getVolume() : int
+ getCurrentChannelNumber() : int
+ receiveBroadcast() : String
+ programChannel(number : int, c : Channel)
+ toString() : String
- increaseVolume()
- decreaseVolume()
- upOneChannel()
- downOneChannel()
- getCurrentChannel() : Channel
- dispatchCommand(c : String)
- turnOn()
- turnOff()
Channel
+ GAUSSIAN_NOISE : String
- s : Scanner
- name : String
+ Channel(network : String) «constructor»
+ stream() : String
+ getName() : String
+ toString() : String
LivingRoom
+ main(args : String[]) «main»
-tv
Remote
- button1 : String = null
- button2 : String = null
- button3 : String = null
- button4 : String = null
- button5 : String = null
- button6 : String = null
- button7 : String = null
- tv : Television
+ Remote(tv : Television) «constructor»
+ programButton(buttonNumber : int, command : String)
+ getButtonCommandScript(buttonNumber : int) : String
+ toString() : String

Class Remote

java.lang.Object
    Remote

```
public class Remote
extends java.lang.Object
```

This class represents a television remote for controlling a tv. It is designed to operate with the `Television` as controller that issues commands.

Constructor Summary

Constructors

Constructor	Description
`Remote(Television tv)`	Pairs the `Remote` object to a particular `Television`

Method Summary

All Methods   Instance Methods   Concrete Methods

Modifier and Type	Method	Description
`java.lang.String`	`getButtonCommandScript(int buttonNumber)`	
`void`	`press(int buttonNumber)`	Issues a command to the paired `Television`.
`void`	`programButton(int buttonNumber, java.lang.String command)`	Sets a particular button to contain a particular `String`.
`java.lang.String`	`toString()`	

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructor Details

Remote

```
public Remote(Television tv)
```

Pairs the `Remote` object to a particular `Television`

Parameters:

`tv` - the television that the remote controls

Method Details

press

```
public void press(int buttonNumber)
```

Issues a command to the paired `Television`. What command is issued depends upon which button is pushed and what that button has been programmed to do.

Parameters:

`buttonNumber` - the number of the button that is to be pressed

programButton

```
public void programButton(int buttonNumber,
                          java.lang.String command)
```

Sets a particular button to contain a particular `String`. The `String` represents a sequence of commands to be issued to the `Television`

Parameters:

`buttonNumber` - the button to be programmed

`command` - represents a sequence of television commands that will happen when that button is pressed

getButtonCommandScript

```
public java.lang.String getButtonCommandScript(int buttonNumber)
```

Parameters:

`buttonNumber` - the button whose command script will be returned

Returns:

returns the command script for a particular button

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

Returns:

returns a catalog of the command script for each button of the remote

Class Channel

java.lang.Object
    Channel

```
public class Channel
extends java.lang.Object
```

Represents a channel that a television can tune to and receive a broadcast stream from. The broadcast stream or television station is represented by a text file that the channel consumes.

Field Summary

Fields

Modifier and Type	Field	Description
`static java.lang.String`	`GAUSSIAN_NOISE`	A constant representing the television fuzz that is returned before a channel is programmed onto a television

Constructor Summary

Constructors

Constructor	Description
`Channel(java.lang.String network)`	Creates a new `Channel` and gives it a network name.

Method Summary

All Methods   Instance Methods   Concrete Methods

Modifier and Type	Method	Description
`java.lang.String`	`getName()`	
`java.lang.String`	`stream()`	
`java.lang.String`	`toString()`	

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Details

GAUSSIAN_NOISE

```
public static java.lang.String GAUSSIAN_NOISE
```

A constant representing the television fuzz that is returned before a channel is programmed onto a television

Constructor Details

Channel

```
public Channel(java.lang.String network)
        throws java.io.IOException
```

Creates a new `Channel` and gives it a network name. The network name is used to link to a file of the same name from which the channel will stream content.

Parameters:

`network` - the name of a tv station that will be used to find a file of the same name from which content will be delivered

Throws:

`java.io.IOException`

Method Details

stream

```
public java.lang.String stream()
```

Returns:

returns content from a broadcast network pulled from the network's file. If there is no content left, `GAUSSIAN_NOISE` is returned

getName

```
public java.lang.String getName()
```

Returns:

returns the name of the network the channel streams from

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

Returns:

also returns the name of the network the channel streams from

Class Television

java.lang.Object
    Television

```
public class Television
extends java.lang.Object
```

Represents an old-fashioned television that receives content live from an airwave broadcast

Field Summary

Fields

Modifier and Type	Field	Description
`static java.lang.String`	`DECREASE_VOLUME`	The commmand to decrease the volume one level
`static java.lang.String`	`DOWN_ONE_CHANNEL`	The command to move down one channel on the television
`static int`	`HIGHEST_CHANNEL`	The highest channel available on the tv
`static int`	`HIGHEST_VOLUME`	The highest volume the television can be set to
`static java.lang.String`	`INCREASE_VOLUME`	The command to increase the volume one level
`static int`	`LOWEST_CHANNEL`	The lowest channel available on the tv
`static int`	`LOWEST_VOLUME`	The lowest volume the television can be set to
`static java.lang.String`	`TURN_OFF`	The command to turn the television off
`static java.lang.String`	`TURN_ON`	The command to turn the television on
`static java.lang.String`	`UP_ONE_CHANNEL`	The command to move up one channel on the television

Constructor Summary

Constructors

Constructor	Description
`Television()`	Begins life off with its channel set to 1, its volume set to 5, and none of its channels programmed to a network

Method Summary

All Methods Instance Methods Concrete Methods

Modifier and Type	Method	Description
`int`	`getCurrentChannelNumber()`	
`java.lang.String`	`getCurrentNetwork()`	
`int`	`getVolume()`	
`void`	`programChannel(int number, Channel c)`	Sets one of the television's channels to a particular channel
`java.lang.String`	`receiveBroadcast()`	
`void`	`receiveCommand(java.lang.String c)`	Issues a sequence of commands to the television.
`java.lang.String`	`toString()`	

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Field Details

TURN_ON

```
public static final java.lang.String TURN_ON
```

The command to turn the television on

See Also:
Constant Field Values

TURN_OFF

```
public static final java.lang.String TURN_OFF
```

The command to turn the television off

See Also:
Constant Field Values

UP_ONE_CHANNEL

```
public static final java.lang.String UP_ONE_CHANNEL
```

The command to move up one channel on the television

See Also:
Constant Field Values

DOWN_ONE_CHANNEL

```
public static final java.lang.String DOWN_ONE_CHANNEL
```

The command to move down one channel on the television

See Also:
Constant Field Values

INCREASE_VOLUME

```
public static final java.lang.String INCREASE_VOLUME
```

The command to increase the volume one level

See Also:
Constant Field Values

DECREASE_VOLUME

```
public static final java.lang.String DECREASE_VOLUME
```

The commmand to decrease the volume one level

See Also:

HIGHEST_CHANNEL

```
public static final int HIGHEST_CHANNEL
```

The highest channel available on the tv

See Also:

Constant Field Values

LOWEST_VOLUME

```
public static final int LOWEST_VOLUME
```

The lowest volume the television can be set to

See Also:

Constant Field Values

HIGHEST_VOLUME

```
public static final int HIGHEST_VOLUME
```

The highest volume the television can be set to

See Also:

Constant Field Values

Constructor Details

Television

```
public Television()
```

Begins life off with its channel set to 1, its volume set to 5, and none of its channels programmed to a network

Method Details

programChannel

```
public void programChannel(int number,
                           Channel c)
```

Sets one of the television's channels to a particular channel

Parameters:

`number` - the channel number to be programmed

`c` - the actual channel from which content will be streamed

receiveBroadcast

```
public java.lang.String receiveBroadcast()
```

receiveCommand

```
public void receiveCommand(java.lang.String c)
```

Issues a sequence of commands to the television. Each command is separated by a space and issued individually one at a time from left to right.

Parameters:

`c` - the sequence of commands the television will be asked to execute

getVolume

```
public int getVolume()
```

Returns:

returns the current volume level of the television

getCurrentChannelNumber

```
public int getCurrentChannelNumber()
```

Returns:

returns the number of the current channel

getCurrentNetwork

```
public java.lang.String getCurrentNetwork()
```

Returns:

returns the name of the current channel's network

toString

```
public java.lang.String toString()
```

Overrides:

`toString` in class `java.lang.Object`

Returns:

returns whether the set is on or off, information about the current channel, and the volume level

```java
import java.util.Scanner;
import java.io.File;
import java.io.IOException;

public class Channel {
  public Channel(String network) throws IOException{
   this.name = network;
   s = new Scanner(new File(network + ".txt"));
  }

  public String stream(){
    String screen = GAUSSIAN_NOISE;

    if(s.hasNext()){
      screen = s.nextLine();
    }

    return screen;
  }

  public String getName(){
    return name;
  }

  public String toString(){
    return getName();
  }

  public static String GAUSSIAN_NOISE = "<GAUSSIAN STATIC>.34.2.92-23-92-935.25029sjlkgset39690)(%*)(^*@#(@)(^)(@#%)TLj..c...gdkgkjp[]
[ppppfj4309506390346t#%*(@)@(%(@)// fjkalsf;lsksjflkslskkdfjsk...<NO BROADCAST>";
  private Scanner s;
  private String name;
}
```

```java
import java.io.IOException;

public class LivingRoom{
  public static void main(String [] args) throws IOException{
    Television tv = new Television();
    Remote remote = new Remote(tv);

    Channel nbc = new Channel("nbc");
    Channel mtv = new Channel("mtv");
    Channel abc = new Channel("abc");

    tv.programChannel(1, nbc);
    tv.programChannel(3, mtv);

    remote.programButton(1, Television.TURN_ON);
    remote.programButton(2, Television.TURN_OFF);
    remote.programButton(3, Television.UP_ONE_CHANNEL);
    remote.programButton(4, Television.DOWN_ONE_CHANNEL);
    remote.programButton(5, Television.INCREASE_VOLUME);
    remote.programButton(6, Television.DECREASE_VOLUME);


    String maxVolumeCommandScript = "";

    for(int i = 0; i < Television.HIGHEST_VOLUME; i++){
      maxVolumeCommandScript += (Television.INCREASE_VOLUME + " ");
    }

    remote.programButton(7, maxVolumeCommandScript);
    System.out.println(remote.getButtonCommandScript(7));

    String muteCommandScript = "";

    for(int i = 0; i < Television.HIGHEST_VOLUME; i++){
      muteCommandScript += (Television.DECREASE_VOLUME + " ");
    }

    remote.programButton(8, muteCommandScript);
    System.out.println(remote.getButtonCommandScript(8));

    System.out.println(tv);

    remote.press(1);
    remote.press(3);
    remote.press(3);
    remote.press(7);

    System.out.println(tv);

    System.out.println(tv.receiveBroadcast());
    System.out.println(tv.receiveBroadcast());

    remote.press(3);
    remote.press(3);
    remote.press(3);

    System.out.println(tv);

    System.out.println(tv.receiveBroadcast());
    System.out.println(tv.receiveBroadcast());

    remote.press(3);
    System.out.println(tv);
    System.out.println(tv.receiveBroadcast());
  }
}
```

```
public class Remote{
  public Remote(Television tv){
    this.tv = tv;
  }

  public void press(int buttonNumber){
    switch(buttonNumber){
      case 1:
        tv.receiveCommand(button1);
          break;
      case 2:
          tv.receiveCommand(button2);
          break;
      case 3:
          tv.receiveCommand(button3);
          break;
      case 4:
          tv.receiveCommand(button4);
          break;
      case 5:
          tv.receiveCommand(button5);
          break;
      case 6:
          tv.receiveCommand(button6);
          break;
      case 7:
          tv.receiveCommand(button7);
          break;
      case 8:
          tv.receiveCommand(button8);
          break;
    }
  }

  public void programButton(int buttonNumber, String command){
    switch(buttonNumber){
      case 1:
        button1 = command;
          break;
      case 2:
          button2 = command;
          break;
      case 3:
          button3 = command;
          break;
      case 4:
          button4 = command;
          break;
      case 5:
          button5 = command;
          break;
      case 6:
          button6 = command;
          break;
      case 7:
          button7 = command;
          break;
      case 8:
          button8 = command;
          break;
    }
  }
  public String getButtonCommandScript(int buttonNumber){
      String commandScript = "";

      switch(buttonNumber){
        case 1:
          commandScript = button1;
          break;
        case 2:
          commandScript = button2;
          break;
        case 3:
          commandScript = button3;
          break;
        case 4:
          commandScript = button4;
          break;
        case 5:
          commandScript = button5;
          break;
        case 6:
          commandScript = button6;
          break;
        case 7:
          commandScript = button7;
          break;
        case 8:
          commandScript = button8;
          break;
      }

      return commandScript;
  }


  public String toString(){
    String strRemote = "";

    strRemote += ("button1: " + button1);
    strRemote += ("button2: " + button2);
    strRemote += ("button3: " + button3);
    strRemote += ("button4: " + button4);
    strRemote += ("button5: " + button5);
    strRemote += ("button6: " + button6);
    strRemote += ("button7: " + button7);
    strRemote += ("button8: " + button8);

    return strRemote;
  }

  private String button1;
  private String button2;
  private String button3;
  private String button4;
  private String button5;
  private String button6;
  private String button7;
  private String button8;

  private Television tv;
}
```

```
public class **Television**{

  public Television(){
    isOn = false;
    currentChannel = 1;
    volume = 5;
  }

  public void programChannel(int number, Channel c){
    switch(number){
      case 1:
        c1 = c;
        break;
      case 2:
          c2 = c;
          break;
      case 3:
          c3 = c;
        break;
      case 4:
          c4 = c;
          break;
      case 5:
          c5 = c;
          break;
      default:
          c5 = c;
    }
  }

  private void increaseVolume(){
   if(volume < HIGHEST_VOLUME){
     volume++;
   }
  }

  private void decreaseVolume(){
    if(volume > LOWEST_VOLUME){
      volume--;
    }
  }

  private void upOneChannel(){
    if(currentChannel < HIGHEST_CHANNEL){
      currentChannel++;
    } else {
      currentChannel = LOWEST_CHANNEL;
    }
  }

  private void downOneChannel(){
    if(currentChannel > LOWEST_CHANNEL){
      currentChannel--;
    } else {
      currentChannel = HIGHEST_CHANNEL;
    }
  }

  public String receiveBroadcast(){
    String stream = "<Tv is off>";

    if(isOn){
      if(getCurrentChannel() != null){
        stream = getCurrentChannel().stream();
      } else{
        stream = Channel.GAUSSIAN_NOISE;
      }
    }

    return stream;
  }

  private Channel getCurrentChannel(){

    Channel channelToReturn = null;

    switch(currentChannel){
      case 1:
        channelToReturn = c1;
          break;
      case 2:
          channelToReturn = c2;
          break;
      case 3:
          channelToReturn = c3;
          break;
      case 4:
          channelToReturn = c4;
          break;
      case 5:
          channelToReturn = c5;
          break;
    }

    return channelToReturn;
  }

  private void dispatchCommand(String c){
    switch(c){
      case TURN_ON:
          turnOn();
          break;
      case TURN_OFF:
          turnOff();
          break;
      case UP_ONE_CHANNEL:
          upOneChannel();
          break;
      case DOWN_ONE_CHANNEL:
          downOneChannel();
          break;
      case INCREASE_VOLUME:
          increaseVolume();
          break;
      case DECREASE_VOLUME:
          decreaseVolume();
          break;
    }
  }

  private void turnOn(){
    isOn = true;
  }

  private void turnOff(){
    isOn = false;
  }

  public void receiveCommand(String c){
    if(isOn){

      String [] splitted = c.split(" ");

      for(int i = 0; i < splitted.length; i++){
        dispatchCommand(splitted[i]);
      }

    } else{

     if(c.equals("turnOn")){
       dispatchCommand(c);
     }

    }
  }

  public int getVolume(){
    return volume;
  }

  public int getCurrentChannelNumber(){
    return currentChannel;
  }

  public String getCurrentNetwork(){
    return getCurrentChannel().getName();
  }


  public String toString(){
    String str = "";

    str += ("\nSet: " + (isOn ? "On" : "Off") + "\n");
    str += ("Channel " + getCurrentChannelNumber() + ": " +
getCurrentChannel() + "\n");
    str += ("Volume: " + getVolume() + "\n\n");

    return str;
  }

  public static final String TURN_ON = "turnOn";
  public static final String TURN_OFF = "turnOff";
  public static final String UP_ONE_CHANNEL = "upOneChannel";
  public static final String DOWN_ONE_CHANNEL = "downOneChannel";
  public static final String INCREASE_VOLUME = "increaseVolume";
  public static final String DECREASE_VOLUME = "decreaseVolume";

  public static final int LOWEST_CHANNEL = 1;
  public static final int HIGHEST_CHANNEL = 5;
  public static final int LOWEST_VOLUME = 0;
  public static final int HIGHEST_VOLUME = 10;

  private Channel c1;
  private Channel c2;
  private Channel c3;
  private Channel c4;
  private Channel c5;

  private boolean isOn;
  private int volume;
  private int currentChannel;

}
```

Text Files

Each block is its own file,
the name of the file at the top of each block is not part of the file

vh1.txt:

```
vh1 signal 1
vh1 signal 2
vh1 signal 3
vh1 signal 4
vh1 signal 5
vh1 signal 6
vh1 signal 7
vh1 signal 8
vh1 signal 9
vh1 signal 10
```

cmt.txt:

```
cmt signal 1
cmt signal 2
cmt signal 3
cmt signal 4
cmt signal 5
cmt signal 6
cmt signal 7
cmt signal 8
cmt signal 9
cmt signal 10
```

cnn.txt:

```
cnn signal 1
cnn signal 2
cnn signal 3
cnn signal 4
cnn signal 5
cnn signal 6
cnn signal 7
cnn signal 8
cnn signal 9
cnn signal 10
```

cn.txt:

```
cn signal 1
cn signal 2
cn signal 3
cn signal 4
cn signal 5
cn signal 6
cn signal 7
cn signal 8
cn signal 9
cn signal 10
```

hbo.txt:

```
hbo signal 1
hbo signal 2
hbo signal 3
hbo signal 4
hbo signal 5
hbo signal 6
hbo signal 7
hbo signal 8
hbo signal 9
hbo signal 10
```

tbs.txt:

```
tbs signal 1
tbs signal 2
tbs signal 3
tbs signal 4
tbs signal 5
tbs signal 6
tbs signal 7
tbs signal 8
tbs signal 9
tbs signal 10
```

mtv.txt:

```
mtv signal 1
mtv signal 2
mtv signal 3
mtv signal 4
mtv signal 5
mtv signal 6
mtv signal 7
mtv signal 8
mtv signal 9
mtv signal 10
```

nbc.txt:

```
nbc signal 1
nbc signal 2
nbc signal 3
nbc signal 4
nbc signal 5
nbc signal 6
nbc signal 7
nbc signal 8
nbc signal 9
nbc signal 10
```

qvc.txt:

```
qvc signal 1
qvc signal 2
qvc signal 3
qvc signal 4
qvc signal 5
qvc signal 6
qvc signal 7
qvc signal 8
qvc signal 9
qvc signal 10
```

abc.txt:

```
abc signal 1
abc signal 2
abc signal 3
abc signal 4
abc signal 5
abc signal 6
abc signal 7
abc signal 8
abc signal 9
abc signal 10
```

cbs.txt:

```
cbs signal 1
cbs signal 2
cbs signal 3
cbs signal 4
cbs signal 5
cbs signal 6
cbs signal 7
cbs signal 8
cbs signal 9
cbs signal 10
```

cc.txt:

```
cc signal 1
cc signal 2
cc signal 3
cc signal 4
cc signal 5
cc signal 6
cc signal 7
cc signal 8
cc signal 9
cc signal 10
```