## 一、 实验名称

# Classes and Objects

## 二、 实验目的

- Be able to declare a new class
- Be able to write a constructor
- Be able to write instance methods that return a value
- Be able to write instance methods that take arguments
- Be able to instantiate an object
- Be able to use calls to instance methods to access and change the state of an object

## 三、 实验内容

Task #1 Creating a New Class
Task #2 Writing a Constructor
Task #3 Methods
Task #4 Running the application
Task #5 Creating another instance of a Television

## 四、 实验方法(原理、流程图)

1. Written by Intellij IDEA Community edition 2020.3

2.

Task 1:

（1）In a new file, create a class definition called Television.
（2）Put a program header (comments/documentation) at the top of the file.
（3）Declare the 2 constant fields listed in the UML diagram.
（4）Declare the 3 remaining fields listed in the UML diagram.
（5）Write a comment for each field indicating what it represents.
（6）Save this file as Television.java.

Task 2:

(1) Create a constructor definition that has two parameters, a manufacturer's brand and a screen size. These parameters will bring in information.
(2) Inside the constructor, assign the values taken in from the parameters to the corresponding fields.
(3) Initialize the powerOn field to false (power is off), the volume to 20, and the channel to 2.
(4) Write comments describing the purpose of the constructor above the method header.

Task 3:

(1) Define accessor methods called getVolume, getChannel, getManufacturer, and getScreenSize that return the value of the corresponding field.
(2) Define a mutator method called setChannel accepts a value to be stored in the channel field.

(3) Define a mutator method called power that changes the state from true to false or from false to true.
(4) Define two mutator methods to change the volume.
(5) Write javadoc comments above each method header.

Task 4:

(1) Copy the file TelevisionDemo.java (see code listing 3.1) from the Student CD or as directed by your instructor.
(2) Compile and run TelevisionDemo and follow the prompts.

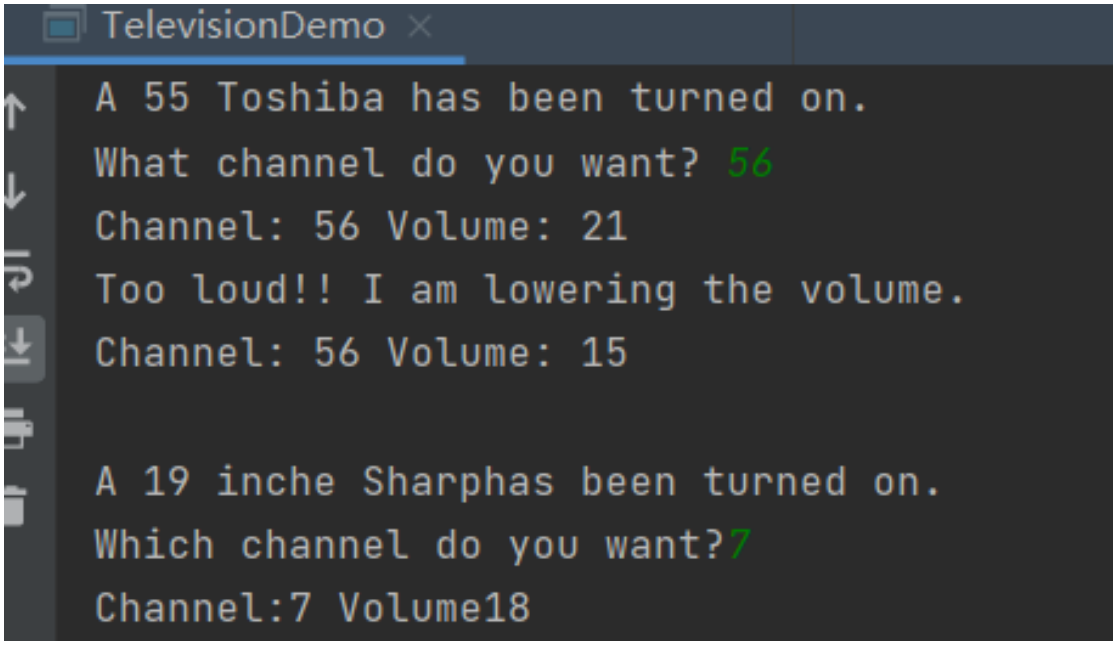
Task 5:

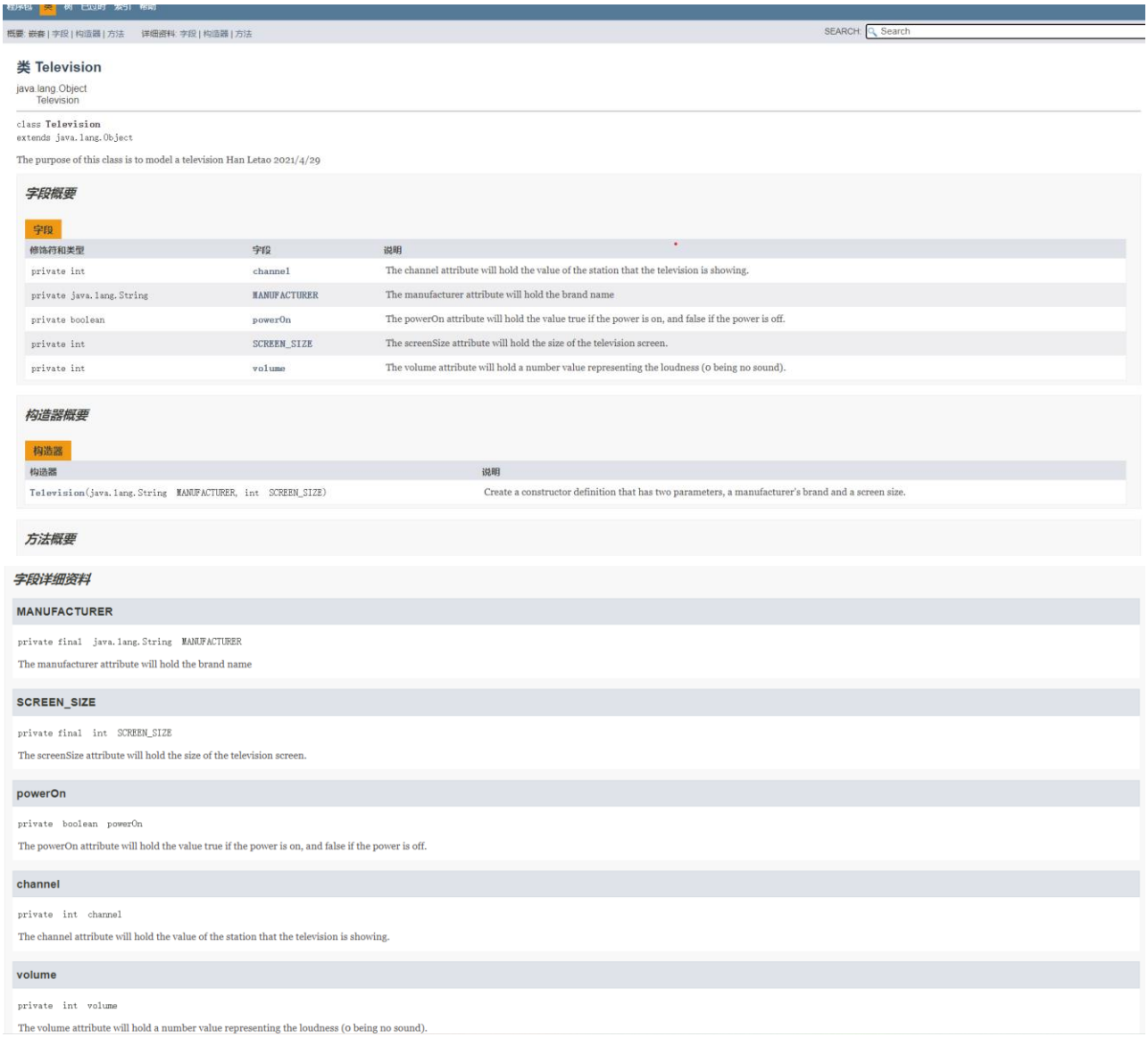
(1) Edit the TelevisionDemo.java file.
(2) Declare another Television object called portable.
(3) Instantiate portable to be a Sharp 19 inch television.
(4) Use a call to the power method to turn the power on.
(5) Use calls to the accessor methods to print what television was turned on.
(6) Use calls to the mutator methods to change the channel to the user's preference and decrease the volume by two.
(7) Use calls to the accessor methods to print the changed state of the portable.
(8) Compile and debug this class.

## 五、 实验结论

The experimental requirements have been successfully realized.

The results of manual calculation are the same as those calculated by my codes.





概要: 嵌套 | 字段 | 构造器 | 方法　详细资料: 字段 | 构造器 | 方法　SEARCH: Search

## 类 Television

java.lang.Object
　Television

class **Television**
extends java.lang.Object

The purpose of this class is to model a television Han Letao 2021/4/29

### 字段概要

字段

| 修饰符和类型 | 字段 | 说明 |
|---|---|---|
| private int | channel | The channel attribute will hold the value of the station that the television is showing. |
| private java.lang.String | MANUFACTURER | The manufacturer attribute will hold the brand name |
| private boolean | powerOn | The powerOn attribute will hold the value true if the power is on, and false if the power is off. |
| private int | SCREEN_SIZE | The screenSize attribute will hold the size of the television screen. |
| private int | volume | The volume attribute will hold a number value representing the loudness (0 being no sound). |

### 构造器概要

构造器

| 构造器 | 说明 |
|---|---|
| Television(java.lang.String MANUFACTURER, int SCREEN_SIZE) | Create a constructor definition that has two parameters, a manufacturer's brand and a screen size. |

### 方法概要

### 字段详细资料

#### MANUFACTURER

private final java.lang.String MANUFACTURER

The manufacturer attribute will hold the brand name

#### SCREEN_SIZE

private final int SCREEN_SIZE

The screenSize attribute will hold the size of the television screen.

#### powerOn

private boolean powerOn

The powerOn attribute will hold the value true if the power is on, and false if the power is off.

#### channel

private int channel

The channel attribute will hold the value of the station that the television is showing.

#### volume

private int volume

The volume attribute will hold a number value representing the loudness (0 being no sound).

## 六、实验体会和收获

1. Through I successfully put classes and objects into practice and write javadoc correctly.By writing these code,I have a deepy realize to classes and objects.Mainly about how to creat a new class and use this class.

2. When writing these codes,i meet some difficults.But I solve these problemss by search relevant information on the Internet and review PPT.

3. By writing this task I am more interested in Java.

## 七、程序代码

### (1) Television.java

```
/**
 * The purpose of this class is to model a television
 * Han Letao 2021/4/29
 */
class Television
{
    /**
     * The manufacturer attribute will hold the brand name
     */
    final private String MANUFACTURER;
    /**
     * The screenSize attribute will hold the size of the television screen.
     */
    final private int SCREEN_SIZE;
    /**
     * The powerOn attribute will hold the value true if the power is on, and false if
the power is off.
     */
    private boolean powerOn;
    /**
     * The channel attribute will hold the value of the station that the television is
showing.
```

```
     */
    private int channel;
    /**
     * The volume attribute will hold a number value representing the loudness (0
being no sound).
     */
    private int volume;
    /**
     * Create a constructor definition that has two parameters, a manufacturer's
brand and a screen size.
     * @param MANUFACTURER Television's manufacturer.
     * @param SCREEN_SIZE Television's screen size.
     */
    public Television(String MANUFACTURER,int SCREEN_SIZE)
    {
        this.MANUFACTURER = MANUFACTURER;
        this.SCREEN_SIZE = SCREEN_SIZE;
        powerOn = false;
        volume = 20;
        channel = 2;
    }
    /**
```

```
     * Get the television's manufacturer.
     * @return MANUFACTURER The television's manufacturer.
     */
    public String getManufacturer()
    {
        return MANUFACTURER;
    }
    /**
     * Get television's volume.
     * @return volume Television's volume.
     */
    public int getVolume()
    {
        return volume;
    }
    /**
     *Get television's channel.
     * @return channel Television's channel.
     */
    public int getChannel()
    {
        return channel;
```

```
    }

    /**
     * Get television's screen size.
     * @return SCREEN_SIZE Television's screen size.
     */
    public int getScreenSize()
    {
        return SCREEN_SIZE;
    }

    /**
     * Set television's channel.
     * @param station Television's channel.
     */
    public void setChannel(int station)
    {
        channel = station;
    }

    /**
     * Changes the power state from true to false or from false to true
     */
    public void power()
    {
```

```
        powerOn = !powerOn;
    }
    /**
     * Increase volume.
     */
    public void increaseVolume()
    {
        volume += 1;
    }
    /**
     * Decrease volume.
     */
    public void decreaseVolume()
    {
        volume -= 1;
    }
}
```

(2) TelevisionDemo.java

```
/**
 * This class demonstrates the Television class
 */
import java.util.Scanner;
```

```
public class TelevisionDemo
{
    public static void main(String[] args)
    {
        //create a Scanner object to read from the keyboard
        Scanner keyboard = new Scanner (System.in);
        //declare variables
        int station; //the user's channel choice

        //declare and instantiate a television object
        Television bigScreen = new Television("Toshiba", 55);
        //turn the power on
        bigScreen.power();
        //display the state of the television
        System.out.println("A " + bigScreen.getScreenSize() +" "+
                bigScreen.getManufacturer() + " has been turned on.");
        //prompt the user for input and store into station
        System.out.print("What channel do you want? ");
        station = keyboard.nextInt();

        //change the channel on the television
        bigScreen.setChannel(station);
```

```
        //increase the volume of the television
        bigScreen.increaseVolume();
        //display the the current channel and volume of the television
        System.out.println("Channel: " + bigScreen.getChannel() +
                " Volume: " + bigScreen.getVolume());
        System.out.println("Too loud!! I am lowering the volume.");
        //decrease the volume of the television
        bigScreen.decreaseVolume();
        bigScreen.decreaseVolume();
        bigScreen.decreaseVolume();
        bigScreen.decreaseVolume();
        bigScreen.decreaseVolume();
        bigScreen.decreaseVolume();
        //display the current channel and volume of the television
        System.out.println("Channel: " + bigScreen.getChannel() +
                " Volume: " + bigScreen.getVolume());
        System.out.println(); //for a blank line
        //HERE IS WHERE YOU DO TASK #5
        Television portable = new Television("Sharp",19);
        portable.power();
        System.out.println("A"+" "+portable.getScreenSize()+" "+"inche"+"
"+portable.getManufacturer()+"has been turned on.");
```

```
            System.out.print("Which channel do you want?");
            station= keyboard.nextInt();
            portable.setChannel(station);
            portable.decreaseVolume();
            portable.decreaseVolume();
            System.out.println("Channel:"+portable.getChannel()+"
"+"Volume"+portable.getVolume());
        }
}
```