Bongo’s Android Developer position test answers.

1.

// Function to check whether two strings

// are an anagram of each other

static boolean areAnagram(String str1, String str2)

{

HashMap<Character, Integer> hmap1

= new HashMap<Character, Integer>();

HashMap<Character, Integer> hmap2

= new HashMap<Character, Integer>();

char arr1[] = str1.toCharArray();

char arr2[] = str2.toCharArray();

// Mapping first string

for (int i = 0; i < arr1.length; i++) {

if (hmap1.get(arr1[i]) == null) {

hmap1.put(arr1[i], 1);

}

else {

Integer c = (int)hmap1.get(arr1[i]);

hmap1.put(arr1[i], ++c);

}

}

// Mapping second String

for (int j = 0; j < arr2.length; j++) {

if (hmap2.get(arr2[j]) == null)

hmap2.put(arr2[j], 1);

else {

Integer d = (int)hmap2.get(arr2[j]);

hmap2.put(arr2[j], ++d);

}

}

if (hmap1.equals(hmap2))

return true;

else

return false;

}

// Test function

public static void test(String str1, String str2)

{

System.out.println("Strings to be checked are:\n"

+ str1 + "\n" + str2 + "\n");

// Find the result

if (areAnagram(str1, str2))

System.out.println("The two strings are"

+ "anagram of each other\n");

else

System.out.println("The two strings are not"

+ " anagram of each other\n");

}

// Driver program

public static void main(String args[])

{

// Get the Strings

String str1 = "geeksforgeeks";

String str2 = "forgeeksgeeks";

// Test the Strings

test(str1, str2);

// Get the Strings

str1 = "geeksforgeeks";

str2 = "geeks";

// Test the Strings

test(str1, str2);

}

2.

Explain the design pattern used in following:

interface Vehicle {

int set\_num\_of\_wheels()

int set\_num\_of\_passengers()

boolean has\_gas()

}

a) Explain how can you use the pattern to create car and plane class?

The design pattern that has been here is factory design pattern.To create a car and a plane class,we will

simply keep Vehicle class as a base class.The key point of factory design pattern is we define a class type/interface and then

we will have number of subclasses which implement the contract defined by the base class.Here the subclasses are

Car and Plane class

For Example-

Class Car implemnts Vehicle{

public int set\_num\_of\_wheels(){

return 4;

}

}

public int set\_num\_of\_passengers(){

return 5;

}

}

public boolean has\_gas(){

return true;

}

}

}

Class Plane implements Vehicle{

public int set\_num\_of\_wheels(){

return 3;

}

}

public int set\_num\_of\_passengers(){

return 150;

}

}

public boolean has\_gas(){

return false;

}

}

}

b) Use a different design pattern for this solution.

I will be going to use Decorator design pattern for this solution.

Decorator design pattern is used to modify the functionality of an object at runtime. At the same

time other instances of the same class will not be affected by this, so individual object gets the

modified behavior. Decorator design pattern is one of the structural design pattern (such as Adapter

Pattern, Bridge Pattern, Composite Pattern) and uses abstract classes or interface with composition

to implement.

This is our interface that we are going to solve with Decorator design pattern

interface Vehicle {

int set\_num\_of\_wheels()

int set\_num\_of\_passengers()

boolean has\_gas()

}

public class BaseVehicle implements Vehicle {

protected Vehicle vehicle;

public BaseVehicle(Vehicle c){

this.vehicle=c;

}

@Override

public int set\_num\_of\_wheels() {

this.vehicle.set\_num\_of\_wheels();

}

@Override

public int set\_num\_of\_passengers() {

this.vehicle.set\_num\_of\_passengers();

}

@Override

public boolean has\_gas() {

this.vehicle.has\_gas();

}

}

Now Extending the base decorator functionality and modifying the component behavior accordingly.

We can have concrete decorator classes as Car and Plane

public class Car extends BaseVehicle {

public Car(Car c) {

super(c);

}

@Override

public int set\_num\_of\_wheels(){

super.set\_num\_of\_wheels();

return 4;

}

@Override

public int set\_num\_of\_passengers(){

super.set\_num\_of\_passengers();

return 5;

}

@Override

public boolean has\_gas(){

super.has\_gas();

return true;

}

}

public class Plane extends BaseVehicle {

public Plane(Plane p) {

super(p);

}

@Override

public int set\_num\_of\_wheels(){

super.set\_num\_of\_wheels();

return 3;

}

@Override

public int set\_num\_of\_passengers(){

super.set\_num\_of\_passengers();

return 150;

}

@Override

public boolean has\_gas(){

super.has\_gas();

return false;

}

}

3. Android Video player sample is been already executed on my repository. Check Video player github link: <https://github.com/filelucker/android/tree/master/videoview_sample>

Pseudo code

public class MainActivity extends ActionBarActivity {

static final String HLS\_STREAMING\_SAMPLE = "http://devimages.apple.com/iphone/samples/bipbop/gear1/prog\_index.m3u8";

VideoView sampleVideoView;

MediaController mediaController;

ProgressDialog progressDialog;

SeekBar seekBar;

ImageView playPauseButton, forwardButton, rewindButton;

ImageView stopButton;

TextView runningTime;

int currentPosition;

boolean isRunning = false;

Handler mHandler;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

sampleVideoView = (VideoView) findViewById(R.id.videoView);

sampleVideoView.setVideoURI(Uri.parse(HLS\_STREAMING\_SAMPLE));

playPauseButton = (ImageView) findViewById(R.id.playPauseButton);

playPauseButton.setOnClickListener(playPauseClickListener);

forwardButton = (ImageView) findViewById(R.id.forwardButton);

forwardButton.setOnClickListener(forwardButtonClickListener);

forwardButton.setEnabled(false);

rewindButton = (ImageView) findViewById(R.id.rewindButton);

rewindButton.setOnClickListener(rewindButtonClickListener);

rewindButton.setEnabled(false);

stopButton = (ImageView) findViewById(R.id.stopButton);

stopButton.setOnClickListener(stopClickListener);

seekBar = (SeekBar) findViewById(R.id.seekBar);

seekBar.setOnSeekBarChangeListener(seekBarChangeListener);

runningTime = (TextView) findViewById(R.id.runningTime);

runningTime.setText("00:00");

progressDialog = new ProgressDialog(this);

progressDialog.setTitle("SampleVideoView");

progressDialog.setMessage("Buffering...");

progressDialog.show();

sampleVideoView.setOnPreparedListener(prepareMediaPlayer);

sampleVideoView.setOnCompletionListener(new MediaPlayer.OnCompletionListener() {

@Override

public void onCompletion(MediaPlayer mp) {

Toast.makeText(getBaseContext(), "Play finished", Toast.LENGTH\_LONG).show();

}

});

}

private MediaPlayer.OnPreparedListener prepareMediaPlayer = new MediaPlayer.OnPreparedListener() {

@Override

public void onPrepared(MediaPlayer mp) {

progressDialog.dismiss();

seekBar.setMax(sampleVideoView.getDuration());

sampleVideoView.start();

sampleVideoView.postDelayed(refreshTime, 1000);

playPauseButton.setImageResource(R.mipmap.pause\_button);

rewindButton.setEnabled(true);

forwardButton.setEnabled(true);

}

};

private View.OnClickListener stopClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

playPauseButton.setImageResource(R.mipmap.play\_button);

sampleVideoView.stopPlayback();

currentPosition = 0;

rewindButton.setEnabled(false);

forwardButton.setEnabled(false);

}

};

private View.OnClickListener forwardButtonClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

sampleVideoView.resume();

currentPosition = currentPosition + 5000;

sampleVideoView.seekTo(currentPosition);

playPauseButton.setImageResource(R.mipmap.pause\_button);

}

};

private View.OnClickListener rewindButtonClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

sampleVideoView.resume();

currentPosition = currentPosition - 5000;

sampleVideoView.seekTo(currentPosition);

playPauseButton.setImageResource(R.mipmap.pause\_button);

}

};

private View.OnClickListener playPauseClickListener = new View.OnClickListener() {

@Override

public void onClick(View v) {

if (v.getId() == R.id.playPauseButton) {

//Play video

if (!isRunning) {

isRunning = true;

sampleVideoView.resume();

sampleVideoView.seekTo(currentPosition);

playPauseButton.setImageResource(R.mipmap.pause\_button);

rewindButton.setEnabled(true);

forwardButton.setEnabled(true);

}

//Pause video

else {

isRunning = false;

sampleVideoView.pause();

currentPosition = sampleVideoView.getCurrentPosition();

playPauseButton.setImageResource(R.mipmap.play\_button);

rewindButton.setEnabled(false);

forwardButton.setEnabled(false);

}

}

}

};

private SeekBar.OnSeekBarChangeListener seekBarChangeListener = new SeekBar.OnSeekBarChangeListener() {

@Override

public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {

}

@Override

public void onStartTrackingTouch(SeekBar seekBar) {

}

@Override

public void onStopTrackingTouch(SeekBar seekBar) {

sampleVideoView.seekTo(seekBar.getProgress());

}

};

private Runnable refreshTime = new Runnable() {

@Override

public void run() {

seekBar.setProgress(sampleVideoView.getCurrentPosition());

if (sampleVideoView.isPlaying())

sampleVideoView.postDelayed(refreshTime, 1000);

int time = sampleVideoView.getCurrentPosition() / 1000;

int minute = time / 60;

int second = time % 60;

runningTime.setText(minute + ":" + second);

}

};

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

// Handle action bar item clicks here. The action bar will

// automatically handle clicks on the Home/Up button, so long

// as you specify a parent activity in AndroidManifest.xml.

int id = item.getItemId();

//noinspection SimplifiableIfStatement

if (id == R.id.action\_settings) {

return true;

}

return super.onOptionsItemSelected(item);

}

}