

Μέθοδος επίλυσης ασκήσεων εξετάσεων

Οι εξετάσεις αποτελούνται από 2 τύπους ασκήσεων. Η μια είναι απλή και πιάνει 2 μονάδες και η άλλη πιο σύνθετη που πιάνει 8 μονάδες. Εδώ θα δείτε την μέθοδο επίλυση ασκήσεων.

Εύκολη άσκηση

Η απλή άσκηση ζητάει να φτιαχτεί ένα activity με ένα απλό EditText (το ονομάζει TextBox) που να δέχεται μια είσοδο από τον χρήστη και ένα κουμπί PRESS ME που να λαμβάνει τα δεδομένα, να τα επεξεργάζεται (αν ζητάει να γίνουν πράξεις) και στην συνέχεια να τα εμφανίζει στην οθόνη είτε στη θέση κάποιου TextView είτε σε Toast.

1. Φτιάχνω το **activity_main.xml** όπως μου το περιγράφουν στην εκφώνηση.

Προσοχή δίνω:

- Χρώμα background, θα έχω επιλεγμένο το background και αναζητώ την λέξη background ή στον κώδικα εισάγω

```
android:background="#fff322"
```

πριν το **tools:context=".MainActivity"**

- Χρώμα για κουμπί, έχω επιλεγμένο το κουμπί και αναζητώ backgroundTint ή στον κώδικα εισάγω

```
android:backgroundTint="#7777cb"
```

- Αλλάζω ΟΛΑ τα id των αντικειμένων που εισάγω σε κάτι που να δηλώνει τι κάνει το αντικείμενο. Αυτά θα τα χρησιμοποιήσω στον κώδικα java.

2. Στο αρχείο της MainActivity.java

- Φτιάχνω τα αντικείμενα EditText και Button

```
private EditText editText;  
private Button button;
```

- Τα συνδέω με την διεπαφή

```
// Text και btn είναι τα id από το xml  
editText = findViewById(R.id.Text);
```

```
button = findViewById(R.id.btn);
```

- Βάζω ένα Listener στο button. Εκεί θα λάβω το κείμενο που έκανε εισαγωγή ο χρήστης καθώς και το μήνυμα που θα εμφανίζεται με το Toast.

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String enteredText = editText.getText().toString();  
  
        // Εμφάνιση του μηνύματος με Toast  
        Toast.makeText(MainActivity.this, "Entered Text: " + enteredText,  
            Toast.LENGTH_SHORT).show();  
    }  
});
```

- Περίπτωση που εισάγει νούμερο και χρειαστεί πράξη, πρέπει να το μετατρέψετε σε Integer ή καλύτερα σε Double και μετά να κάνετε την πράξη.

```
button.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String input = editText.getText().toString();  
        if (input.isEmpty()) {  
            number.setError("Please enter a number");  
        } else {  
            double num = Double.parseDouble(input);  
            double sqrt = (double) Math.sqrt(num);  
  
            Toast.makeText(MainActivity.this, "SQRT(" + num + ") = " + sqrt,  
                Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

- Υπάρχει περίπτωση να ζητηθεί να κάνει την εμφάνιση σε κάποιο TextView. Φροντίστε να το εισάγετε στο activity_main.xml να εισάγετε ένα TextView χωρίς κείμενο και στον κώδικα στο MainActivity.java, αφού το συνδέσετε (**message = findViewById(R.id.Text);**), στο button listener να εισάγετε κώδικα όπως ο παρακάτω

```
button.setOnClickListener(v -> message.setText("My name is: Stathis"));
```

Δύσκολη άσκηση

Και η άσκηση αυτή δεν είναι ιδιαίτερα δύσκολη. Απλά έχει περισσότερα βήματα.

1. Διαβάστε προσεκτικά την εκφώνηση. Από την εκφώνηση θα καταλάβετε:

- Πόσα activity χρειάζονται.
- Πόσα και ποια elements σε κάθε activity.
- Ποιες κλάσεις θα χρειαστεί να φτιάξετε.
- Την μορφή των δεδομένων που σας επιστρέφει η απομακρυσμένη υπηρεσία. Από την μορφή θα προκύψει και μια κλάση. Για την μορφή του JSON θα σας βοηθήσει το Postman. Από το Postman θα δείτε και τον κώδικα του OkHttpHandler (αν και μπορείτε να τον πάρετε σχεδόν έτοιμο παρακάτω).

Εισάγοντας στο Postman την διεύθυνση που θα υπάρχει στην εκφώνηση (πχ <http://192.168.1.104/animals/getMedia.php>) θα λάβουμε ένα αποτέλεσμα JSON του τύπου:

```
{
  "Malco": {
    "grouped_types": "Cat",
    "grouped_voices": "Meow",
    "images": "https://media.istockphoto.com/id/1391912463/photo/photo-of-a-cat-si
  },
  "Mproumpis": {
    "grouped_types": "Parrot",
    "grouped_voices": "Tsiou",
    "images": "https://media.istockphoto.com/id/1127685644/photo/blue-and-yellow-m
  },
  "Rex": {
    "grouped_types": "Dog",
    "grouped_voices": "Bark",
    "images": "https://media.istockphoto.com/id/1482199015/photo/happy-puppy-welsh
  }
}
```

Και μπορούμε να πάρουμε τον κώδικα (δείτε παρακάτω στο OkHttpHandler):

```
OkHttpClient client = new OkHttpClient().newBuilder()
    .build();
MediaType mediaType = MediaType.parse("text/plain");
RequestBody body = RequestBody.create(mediaType, "");
Request request = new Request.Builder()
    .url("http://192.168.1.104/animals/getMedia.php")
    .method("POST", body)
    .build();
Response response = client.newCall(request).execute();
```

Όπου μπορούμε να τον αλλάξουμε ελαφρώς.

2. Απαραίτητες ρυθμίσεις για προσπέραση δικτύου και βιβλιοθήκη εικόνων.

- Ενημέρωση του Android Manifest:

Προσθέστε την άδεια INTERNET στο αρχείο AndroidManifest.xml στο σημείο πριν το **application**:

```
<uses-permission android:name="android.permission.INTERNET" />

<application
    android:allowBackup="true"
```

ενώ τελευταία γραμμή πριν το activity να εμφανίζονται οι γραμμές:

```
    android:networkSecurityConfig="@xml/network_security_config"
    android:usesCleartextTraffic="true">

    <meta-data
        android:name="com.google.android.actions"
        android:resource="@xml/network_security_config" />

    <activity
```

- Δημιουργία του αρχείου **network_security_config.xml**:

Δημιουργήστε ένα νέο φάκελο με όνομα res/xml στον φάκελο του προγράμματος (New- → Folder → XML Resources Folder). Εν συνεχεία, δημιουργήστε ένα αρχείο network_security_config.xml μέσα σε αυτόν τον φάκελο. Το περιεχόμενο του αρχείου θα είναι κάτι παρόμοιο με αυτό (η IP θα είναι αυτή που έχετε στον υπολογιστή σας):

```
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">192.168.1.104</domain>
    </domain-config>
</network-security-config>
```

Αυτό το αρχείο ρυθμίζει την πολιτική ασφαλείας του δικτύου για την εφαρμογή σας.

- Βεβαιωθείτε ότι έχετε προσθέσει τις απαραίτητες εξαρτήσεις στο αρχείο **build.gradle ((Module App))** για να χρησιμοποιήσετε τη βιβλιοθήκη **OkHttp**.

```
dependencies {
    implementation 'com.squareup.okhttp3:okhttp:4.9.0'
    implementation 'androidx.appcompat:appcompat:1.4.1'
```

```
...  
}
```

Δεν ξεχνάμε να επιλέξουμε Sync Now.

- Και για εικόνες για να συμπεριλάβετε τη βιβλιοθήκη Picasso στην εφαρμογή, πρέπει να προσθέσετε την αντίστοιχη εξάρτηση (dependency) στο αρχείο **build.gradle ((Module App))** του project.

- Άνοιξε το αρχείο build.gradle που βρίσκεται στον κύριο φάκελο του project (όχι στο φάκελο app).
- Στην ενότητα dependencies, προσθέστε την ακόλουθη γραμμή για να προσθέσετε την εξάρτηση της βιβλιοθήκης Picasso:

```
implementation 'com.squareup.picasso:picasso:2.5.2'
```

Δεν ξεχνάμε να επιλέξουμε Sync Now.

- Πρέπει να δημιουργήσουμε μια κλάση **OkHttpHandler**(OkHttpHandler.java).

Εδώ είναι ένα παράδειγμα κώδικα που χρησιμοποιεί τη βιβλιοθήκη OkHttp για να κάνει ένα απλό GET αίτημα:

```
package com.example.animals;  
  
import android.os.StrictMode;  
  
import org.json.JSONException;  
import org.json.JSONObject;  
  
import java.util.ArrayList;  
import java.util.Iterator;  
import okhttp3.*;  
  
public class OkHttpHandler {  
  
    public OkHttpHandler() {  
        StrictMode.ThreadPolicy policy = new  
            StrictMode.ThreadPolicy.Builder().permitAll().build();  
        StrictMode.setThreadPolicy(policy);  
    }  
  
    ArrayList<Animal> populateDropDown(String url) throws Exception {  
        ArrayList<Animal> aList = new ArrayList<>();  
        OkHttpClient client = new OkHttpClient().newBuilder().build();  
        RequestBody body = RequestBody.create("", MediaType.parse("text/plain"));  
        Request request = new Request.Builder().url(url).method("POST", body).build();  
        Response response = client.newCall(request).execute();  
        String data = response.body().string();  
    }  
}
```

```

        System.out.println("My Response: " + data);
    try {
        JSONObject json = new JSONObject(data);
        Iterator<String> keys = json.keys();

        // Φτιάχνουμε την λίστα από το αρχείο JSON
        while(keys.hasNext()) {
            String name = keys.next();
            String type = json.getJSONObject(name).getString("grouped_types").
                toString();

            String voice = json.getJSONObject(name).getString("grouped_voices").
                toString();

            String images = json.getJSONObject(name).getString("images").
                toString();

            aList.add(new Animal(name, type, voice, images));
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    // Επιστρέφουμε την λίστα
    return aList;
}
}

```

Αν συναντήσετε να επιστρέφει JSONArray αντί για JSONObject, μπορείτε να χρησιμοποιήσετε το παρακάτω.

```

JSONArray jsonArray = new JSONArray(data);
for (int i = 0; i < jsonArray.length(); i++) {
    JSONObject item = jsonArray.getJSONObject(i);

    String image1 = item.getString("image1");
    String image2 = item.getString("image2");
    String oper1 = item.getString("oper1");
    String oper2 = item.getString("oper2");
    String res = item.getString("res");

    aList.add(new Multi(image1, image2, oper1, oper2, res));
}

```

3. Ξεκινάμε με το πρώτο ερώτημα. Θέλει να δημιουργήσουμε ένα Activity με 2 elements. Το ένα θα είναι είτε RadioGroup που θα δημιουργείται δυναμικά με RadioButtons είτε Spinner με τόσα στοιχεία όσα έχει ο πίνακας που λάβατε από το JSON αρχείο και σίγουρα ένα κουμπί που θα σε μεταφέρει στην επόμενη οθόνη (activity).

Στην περίπτωση που είναι RadioGroup ένας τυπικός κώδικας του activity_main.xml είναι:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/schemas.android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/nextButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="120dp"
        android:text="Next"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"/>

    <RadioGroup
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toTopOf="@+id/buttonNext"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

4. Τώρα πάμε να γεμίσουμε το αρχείο **MainActivity.java**. Εδώ πρέπει να συνδέσουμε το RadioGroup και το κουμπί της διεπαφής με τον κώδικα και πρέπει να γεμίσουμε το RadioGroup με τα RadioButtons.

Δηλώνουμε πρώτα την διεύθυνση από την εκφώνηση (λογικά θα είναι η 195.251.211.64), το RadioGroup, την λίστα (το λέει και η εκφώνηση) με την οποία θα γεμίσω το RadioGroup και επίσης τις ιδιότητες που θα έχω βρει από το Postman.

```
private final String IP = "195.251.211.64";
private RadioGroup rg;
private ArrayList animalList;
private String name, type, voice, image;
```

Στην συνέχεια, στην onCreate πρέπει να φτιαχτεί η λίστα (εδώ πχ των ζώων). Για να φτιαχτεί θα πρέπει να ρωτήσω την κλάση ArrayList να μου την φτιάξει (το λέει και η εκφώνηση). Οπότε ξεκινάω με την γραμμή:

```
super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_main);  
animalList = new AnimalList(IP);
```

4. Προτού συνεχίσουμε, καλό είναι να πάμε να φτιάξουμε τις κλάσεις που μας ζητάει. Αυτή είναι η Animal και η AnimalList. Ας ξεκινήσουμε από την απλή:

- Animal.java

Εδώ ορίζουμε τις ιδιότητες και στην συνέχεια τον κατασκευαστή.

Στην συνέχεια υπάρχουν μόνο μέθοδοι get για τις ιδιότητες.

```
import java.util.HashMap;  
  
public class Animal {  
  
    // Ιδιότητες  
    private String type;  
    private String voice;  
    private String name;  
    private String image;  
  
    // Κατασκευαστής  
    public Animal(String name, String type, String voice, String image){  
        this.name = name;  
        this.type = type;  
        this.voice = voice;  
        this.image = image;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public boolean hasName(String name){  
        return this.name.equals(name);  
    }  
  
    public String getType(){  
        return type;  
    }  
  
    public String getVoice(){  
        return voice;  
    }  
  
    public String getImage(){  
        return image;  
    }  
}
```



```
}
```

- AnimalList.java

Εδώ θα φτιάξουμε την λίστα που θα χρησιμοποιήσουμε για να γεμίσουμε το RadioGroup. Επίσης υλοποιούμε μεθόδους get για τις ιδιότητες name, type, voice, image.

```
import java.util.ArrayList;
import java.util.List;

import okhttp3.OkHttpClient;

public class AnimalList {
    // Δηλώνουμε την λίστα που θα χρησιμοποιήσουμε
    private ArrayList<Animal> aList;

    // Κατασκευαστής. Δέχεται μια IP από την Main και την αποθηκεύει ως πλήρη
    // διεύθυνση όπου θα λάβει απαντήσεις.
    public AnimalList(String ip){
        String url = "http://" + ip + "/animals/getMedia.php";

        try {
            // Φτιάχνουμε ένα αντικείμενο OkHttpClientHandler (με το παραπάνω περιεχόμενο)
            OkHttpClientHandler okHttpClientHandler = new OkHttpClientHandler();

            // Αποθηκεύουμε στην λίστα μας το αποτέλεσμα που θα λάβουμε από
            // την δημιουργία από το OkHttpClientHandler και την μέθοδο που έχουμε υλοποιήσει
            aList = okHttpClientHandler.populateDropDown(url);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    // Μέθοδος για να πάρω μια λίστα με όλα τα ονόματα (με αυτήν θα συμπληρώσω
    // το RadioGroup).
    public List<String> getAllNames(){
        List<String> temp = new ArrayList<>();
        for(int i=0; i<aList.size(); i++){
            temp.add(aList.get(i).getName());
        }
        return temp;
    }

    // Μέθοδος για να λάβω τον τύπο του ζώου που θα ζητήσω ανάλογα με το όνομα.
    public String getAnimalType(String name){
        for(int i=0; i<aList.size(); i++){
            if(aList.get(i).hasName(name)){
                return aList.get(i).getType();
            }
        }
    }
}
```

```

        return "nop name";
    }

    // Μέθοδος για να λάβω την φωνή του ζώου που θα ζητήσω ανάλογα με το όνομα.
    public String getAnimalVoice(String name){
        for(int i=0;i<aList.size();i++){
            if(aList.get(i).hasName(name)){
                return aList.get(i).getVoice();
            }
        }
        return "nop voice";
    }

    // Μέθοδος για να λάβω την φωτογραφία του ζώου που θα ζητήσω ανάλογα με το όνομα.
    public String getAnimalImage(String name){
        for(int i=0;i<aList.size();i++){
            if(aList.get(i).hasName(name)){
                return aList.get(i).getImage();
            }
        }
        return "nop Image";
    }
}

```

5. Επιστρέφουμε στο MainActivity.java.

Εδώ αρχικά ενώνουμε την γραφική διεπαφή με τον κώδικα Java.

```

rg = (RadioGroup) findViewById(R.id.radioGroup);
Button nextButton = (Button) findViewById(R.id.nextButton);

```

Στην συνέχεια θα γεμίσουμε το RadioGroup με RadioButtons δυναμικά.

```

// Πρέπει να πάρουμε σε μια λίστα όλα τα ονόματα των ζώων (έχουμε φτιάξει
// μια μέθοδο).
List<String> allAnimals = animalList.getAllNames();

// Ανάλογα με το μέγεθος της λίστας, φτιάχνουμε ένα ένα τα RadioButtons
for(int i=0;i<allAnimals.size();i++) {
    RadioButton rb = new RadioButton(this);
    rb.setText(allAnimals.get(i));
    rb.setId(100+i);
    rg.addView(rb);
}

```

Στην συνέχεια πρέπει να εισάγω έναν Listener (setOnCheckedChangeListener) έτσι ώστε να λάβω την επιλογή του χρήστη και να την αποθηκεύσω στις ιδιότητες που δήλωσα στην αρχή. Αυτές τις ιδιότητες πρέπει να τις περάσω στην επόμενη οθόνη (activity).

```

rg.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup radioGroup, int i) {
        RadioButton rb = (RadioButton) findViewById(i);
        name = rb.getText().toString();
        type = animalList.getAnimalType(name);
        voice = animalList.getAnimalVoice(name);
        image = animalList.getAnimalImage(rb.getText().toString());
    }
});

```

Και αυτά πρέπει να τα εισάγω σε ένα Intent για να τα στείλω στο επόμενο Activity (είναι νέα μέθοδος, εκτός του onCreate).

```

public void sendSecondActivity(View view){
    Intent intent = new Intent(this, SecondActivity.class);
    intent.putExtra("NAME", name);
    intent.putExtra("TYPE", type);
    intent.putExtra("VOICE", voice);
    intent.putExtra("IMAGE", image);
    startActivity(intent);
}

```

Τέλος, μέσα στην onCreate, πρέπει να βάλω και ένα Button Listener που με το πάτημα να ανοίγει το νέο Activity.

```

nextButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        sendSecondActivity(view);
    }
});

```

6. Δημιουργία του SecondActivity.

Επιλέγουμε **New > Activity > Empty views Activity** και δίνουμε το όνομα SecondActivity (θα δημιουργηθεί και ένα xml με όνομα activity_second).

- Ας ξεκινήσουμε από το xml. Από την εκφώνηση θα ζητηθούν 3 elements. Ένα TextView που θα δείχνει κάποιο κείμενο, από κάτω μια φωτογραφία και από κάτω ένα κουμπί BACK για να επιστρέψει στην προηγούμενη οθόνη (activity).

Ας γεμίσουμε το xml πρώτα. Ένας τυπικός κώδικας είναι ο παρακάτω:

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.andro

```

```

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".SecondActivity">

<TextView
    android:id="@+id/message"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView"
    app:layout_constraintBottom_toTopOf="@+id/image"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<ImageView
    android:id="@+id/image"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="120dp"
    app:layout_constraintBottom_toTopOf="@+id/backButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    tools:srcCompat="@tools:sample/avatars" />

<Button
    android:id="@+id/backButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="120dp"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

- Τώρα ξεκινάμε με το SecondActivity.java.

Αρχικά θα ενώσουμε την γραφική διεπαφή με τον κώδικα java. Θα ενώσω το κείμενο, το κουμπί και την εικόνα (μπορώ να κρατήσω τις ίδιες ονομασίες).

```

TextView message = (TextView) findViewById(R.id.message);
ImageView eikona = (ImageView) findViewById(R.id.image);
Button backButton = (Button) findViewById(R.id.backButton);

```

Λαμβάνω το Intent και το χωρίζω στα συστατικά του.

```
Intent intent = getIntent();
String a_name = intent.getStringExtra("NAME");
String a_type = intent.getStringExtra("TYPE");
String a_voice = intent.getStringExtra("VOICE");
String a_image = intent.getStringExtra("IMAGE");
```

Φτιάχνω ένα προσωρινό μήνυμα, όπως το ζητάει η άσκηση.

```
String tempMsg = "My Name is " + a_name + ", I am a " + a_type +
" and I " + a_voice + ".";
```

Στην συνέχεια το βάζω στην γραφική διεπαφή μου.

```
message.setText(tempMsg);
```

Εισάγω και την εικόνα:

```
Picasso.with(getApplicationContext()).load(Uri.parse(a_image)).resize(300,0).
into(eikona);
```

Τέλος, εισάγω τον Listener στο κουμπί για να γυρνάει στην προηγούμενη οθόνη.

```
backButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent i = new Intent(SecondActivity.this, MainActivity.class);
        startActivity(i);
    }
});
```

Αν αντί για RadioGroup ζητήσει Spinner, οι αλλαγές θα γίνουν στο activity_main.xml και στο MainActivity.java.

1. Αν είναι Spinner τότε αλλάζετε τον κώδικα στο activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <Button
```

```

        android:id="@+id/nextButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="120dp"
        android:text="Next"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"/>

<Spinner
    android:id="@+id/spinner"
    android:layout_width="192dp"
    android:layout_height="45dp"
    app:layout_constraintBottom_toTopOf="@+id/nextButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```

2. Στο MainActivity.java θα κάνω τις εξής αλλαγές:

- Δηλώνω αντί για RadioGroup ένα Spinner.

```
private Spinner spinner;
```

- Μέσα στην onCreate θα ενώσω την γραφική διεπαφή xml με το java.

```
spinner = findViewById(R.id.spinner);
```

- Φτιάχνω την λίστα με όλα τα ονόματα.

```
List<String> allNames = animalList.getAllNames();
```

- Πρέπει να φτιαχτεί ένα ArrayAdapter με όλα τα ονόματα και να προσαρμοστεί στο αντικείμενο Spinner.

```

ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_spinner_item, allNames);
spinner.setAdapter(arrayAdapter);

```

- Τέλος πρέπει να μπει ένας Listener που να λαμβάνει την επιλογή του χρήστη.

```
spinner.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    @Override  
    public void onItemClick(AdapterView<?> parentView,  
        View selectedItemView, int position, long id) {  
        String animalName = allNames.get(position);  
        name = animalName;  
        type = animalList.getAnimalType(animalName);  
        voice = animalList.getAnimalVoice(animalName);  
        image = animalList.getAnimalImage(animalName);  
    }  
  
    @Override  
    public void onNothingSelected(AdapterView<?> parentView) {  
        // Do nothing here  
    }  
});
```

Κατά τα άλλα, ο κώδικας είναι ίδιος.