

Bullet Points

- Sort your ideas for activities into top-level activities, category activities, and detail/edit activities. Use the category activities to navigate from the top-level activities to the detail/edit activities.
 - Image resources go in one or more of the drawable* folders. You reference them in your layout using @drawable/image_name. You access them in your activity code using R.drawable.image_name.
 - An ImageView holds an image. Add it to your layout using <ImageView>. Use android:src to set its source, and android:contentDescription to give it an accessible label. The equivalent methods in Java are setImageResource() and setContentDescription().
 - A ListView displays items in a list. Add it to your layout using <ListView>.
 - Use android:entries in your layout to populate the items in your list views from an array defined in strings.xml.
-

Bullet Points

- A ListActivity is an Activity that comes with a ListView. You get a reference to the ListView using getListView().
 - A ListActivity has its own default layout, but you can replace it with your own.
 - An adapter acts as a bridge between an AdapterView and a data source. ListViews and Spinners are both types of AdapterView.
 - An ArrayAdapter is an adapter that works with arrays.
 - Handle click events on Buttons using android:onClick in the layout code.
 - Handle click events on a ListView in a ListActivity by implementing the onListItemClick() method.
 - Handle click events elsewhere by creating a listener and implementing its click event.
-

Display a menu showing all the food you can buy.

Show details of each drink.

Show a list of all our stores.

Display a list of the drinks we sell.

Show details of an item of food.

Display the address and opening times of each store.

Display a start screen with a list of options.

Top Level Activity

Display a start screen with a list of options.



Category Activities

Display
a menu
showing all
the food you
can buy.

Show a list of
all our stores.

Display a list
of the drinks
we sell.

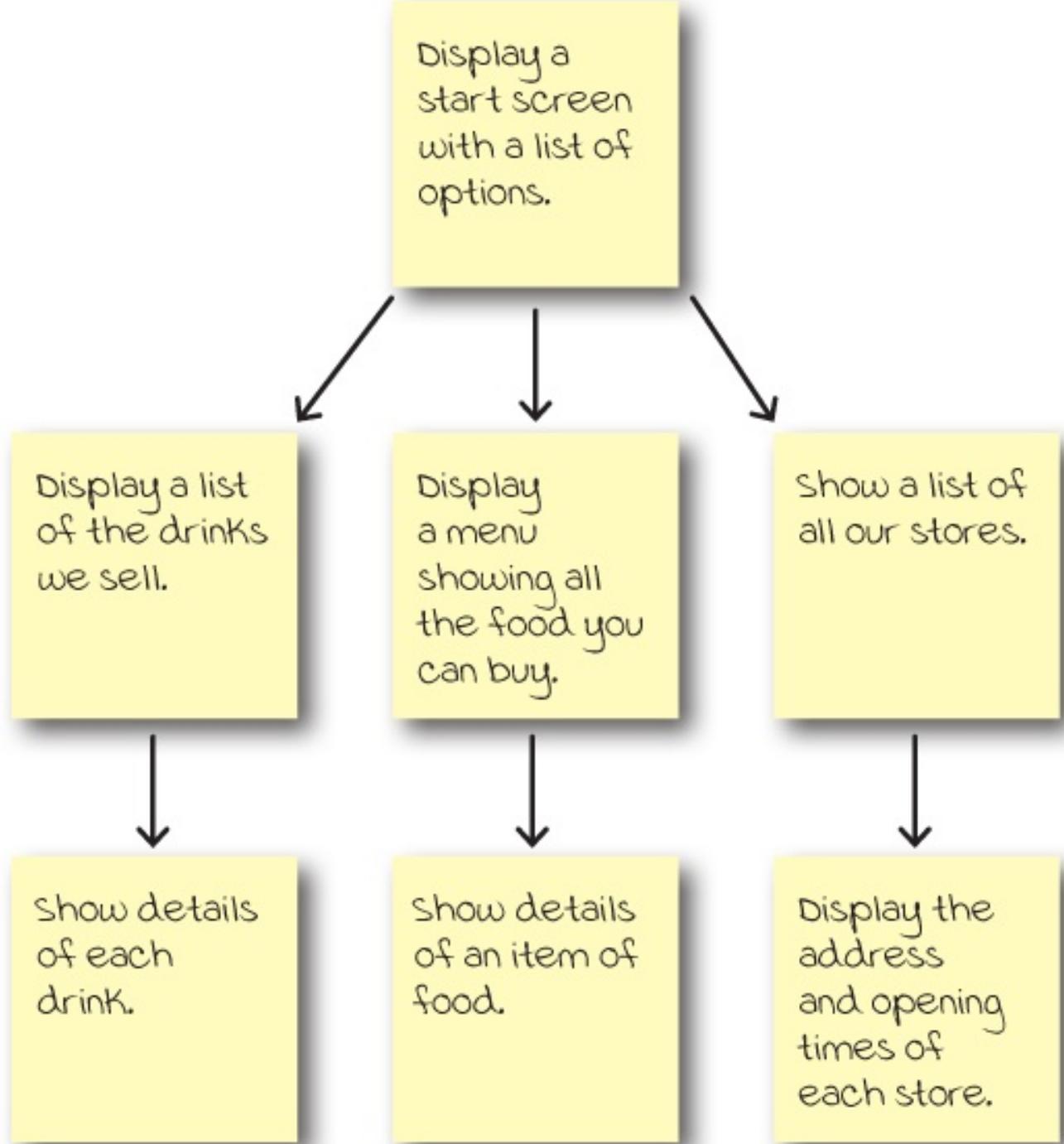
Detail/Edit Activities

show details
of each
drink.

show details
of an item of
food.

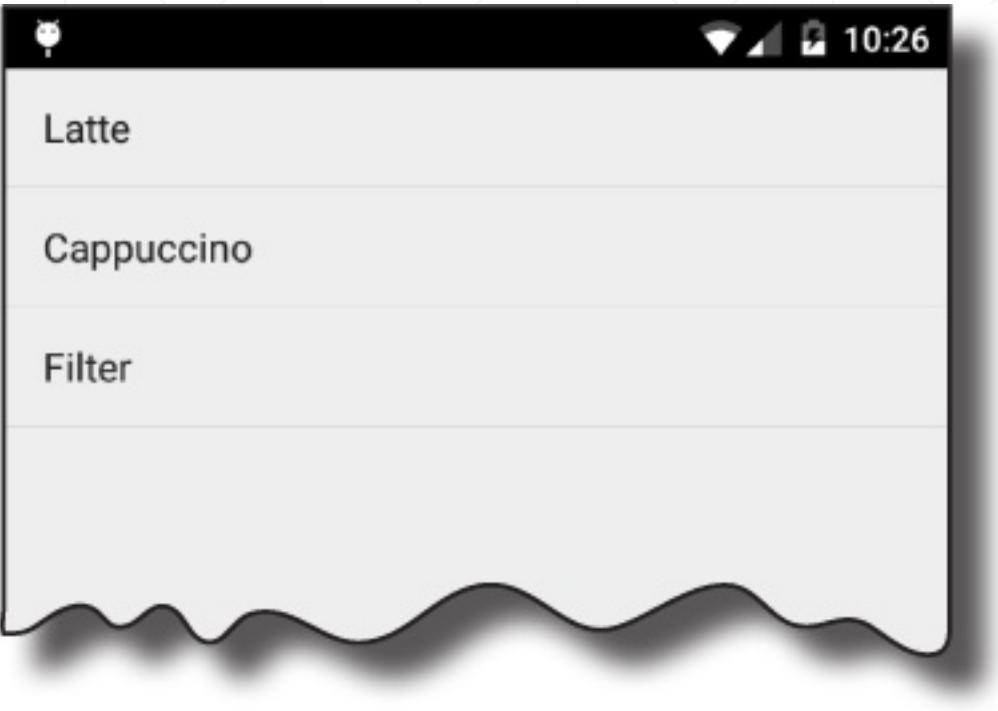
Display the
address
and opening
times of
each store.

Navigating through the Activities



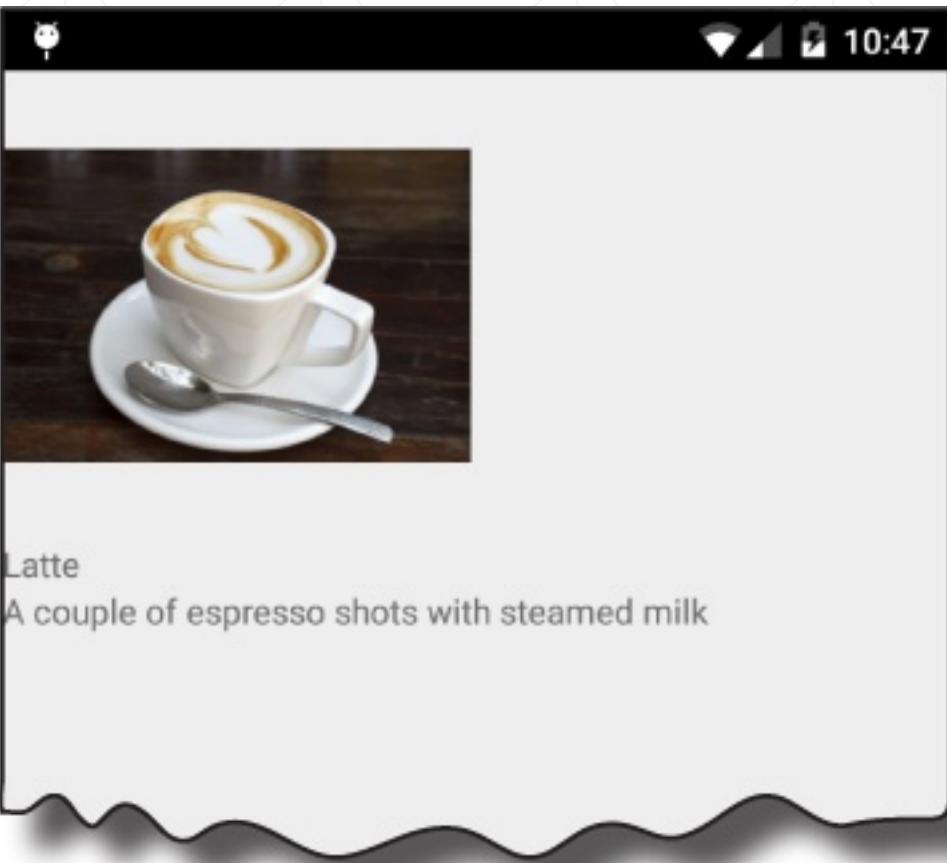
List Views for Categories

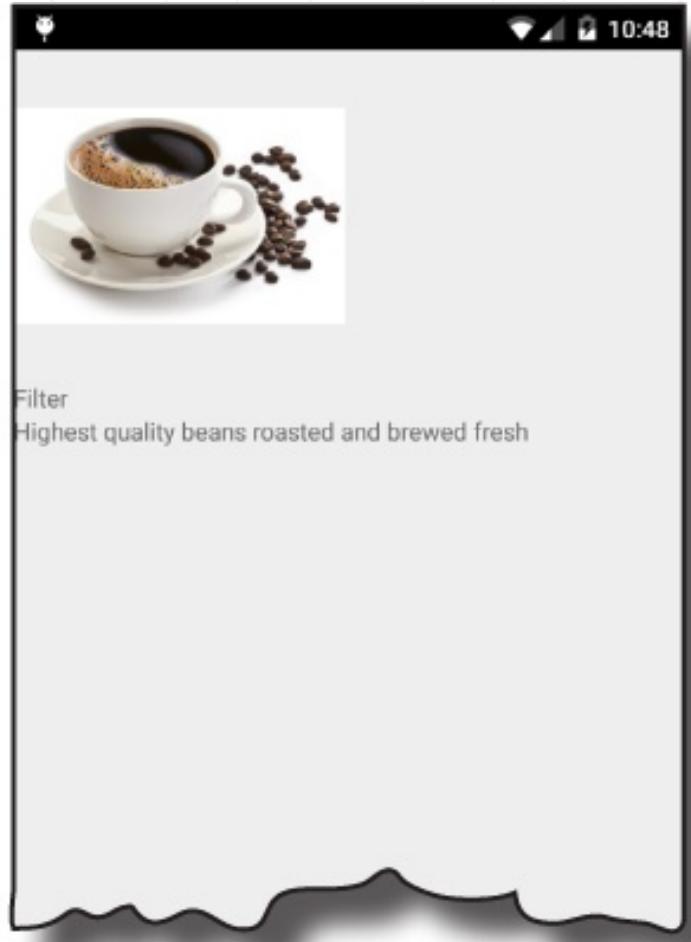
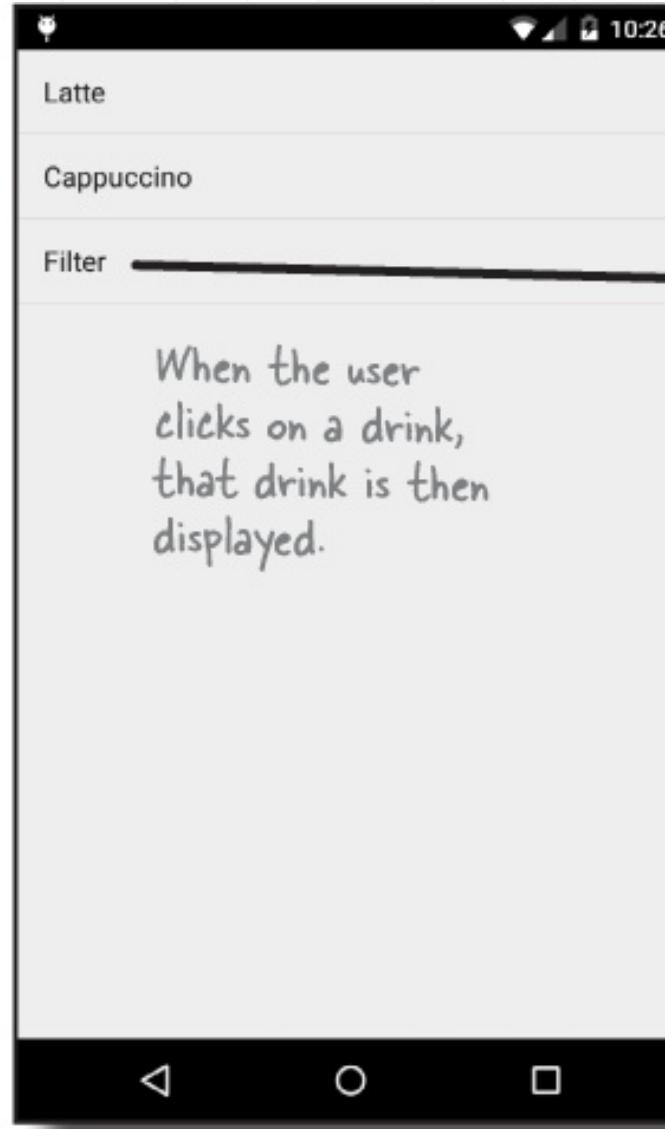
This is a ListView
containing a list of drinks.



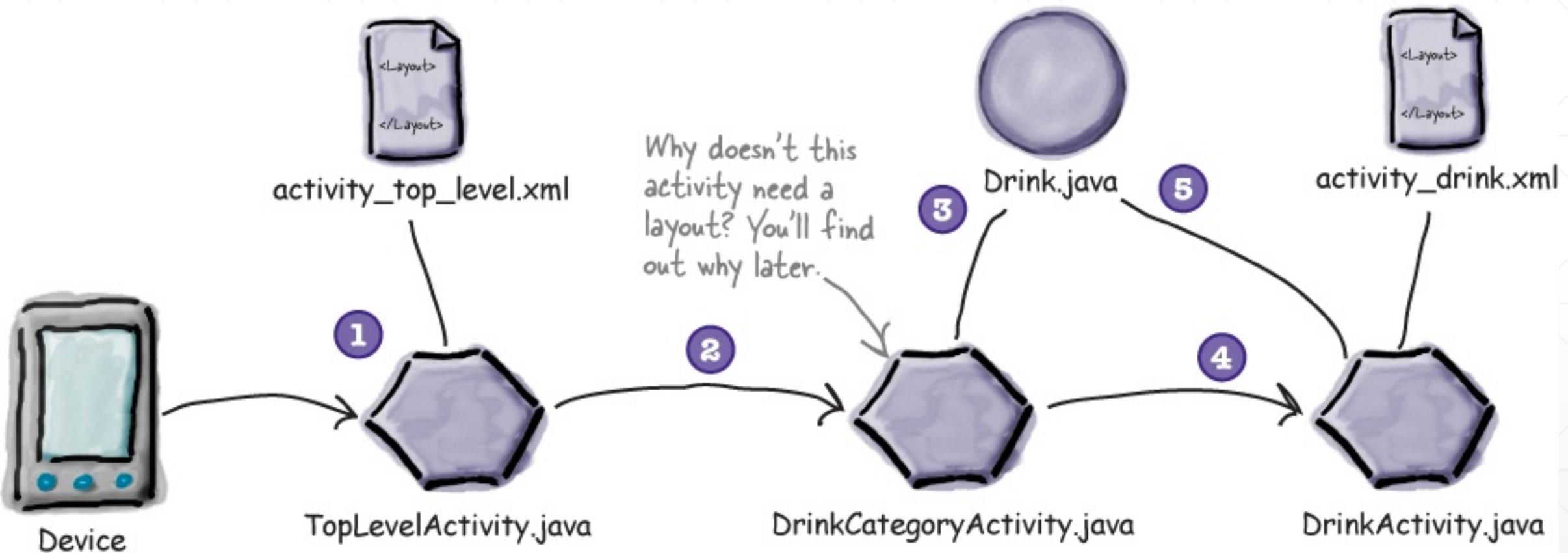
Drink Activity

If you click on the Latte option in the ListView, you get shown the details for the Latte.





Starbuzz App Structure



Steps

- Create Project
 - Add the Drink class and image resources
 - Create TopLevelActivity and its layout
 - Create DrinkCategoryActivity (no layout needed)
 - Create DrinkActivity and it's layout
-

Create New Project

New Project

Android Studio

Configure your new project

Application name: Starbuzz

Company Domain: ltu.edu

Package name: edu.ltu.starbuzz

Project location: /Users/User/Dropbox/VERBATIM/clients/ltu/Starbuzz

Customize the Activity

Creates a new empty activity

Activity Name: TopLevelActivity

Generate Layout File

Layout Name: activity_top_level

Empty Activity

The name of the activity class to create

Cancel Previous Next Finish

```
package com.hfad.starbuzz;

public class Drink {
    private String name;
    private String description;
    private int imageResourceId;
}

//drinks is an array of Drinks
public static final Drink[] drinks = {
    new Drink("Latte", "A couple of espresso shots with steamed milk",
              R.drawable.latte),
    new Drink("Cappuccino", "Espresso, hot milk, and a steamed milk foam",
              R.drawable.cappuccino),
    new Drink("Filter", "Highest quality beans roasted and brewed fresh",
              R.drawable.filter)
};

//Each Drink has a name, description, and an image resource
private Drink(String name, String description, int imageResourceId) {
    this.name = name;
    this.description = description;
    this.imageResourceId = imageResourceId;
}

public String getDescription() {
    return description;
}

public String getName() {
    return name;
}

public int getImageResourceId() {
    return imageResourceId;
}

public String toString() {
    return this.name;
}
}

These are images of the drinks. We'll add these next.
```

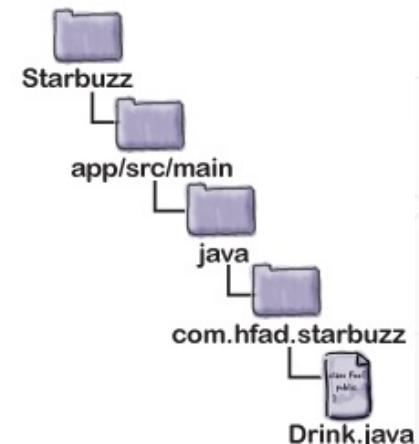
Each Drink has a name, description, and image resource ID. The image resource ID refers to drink images we'll add to the project on the next page.

drinks is an array of three Drinks.

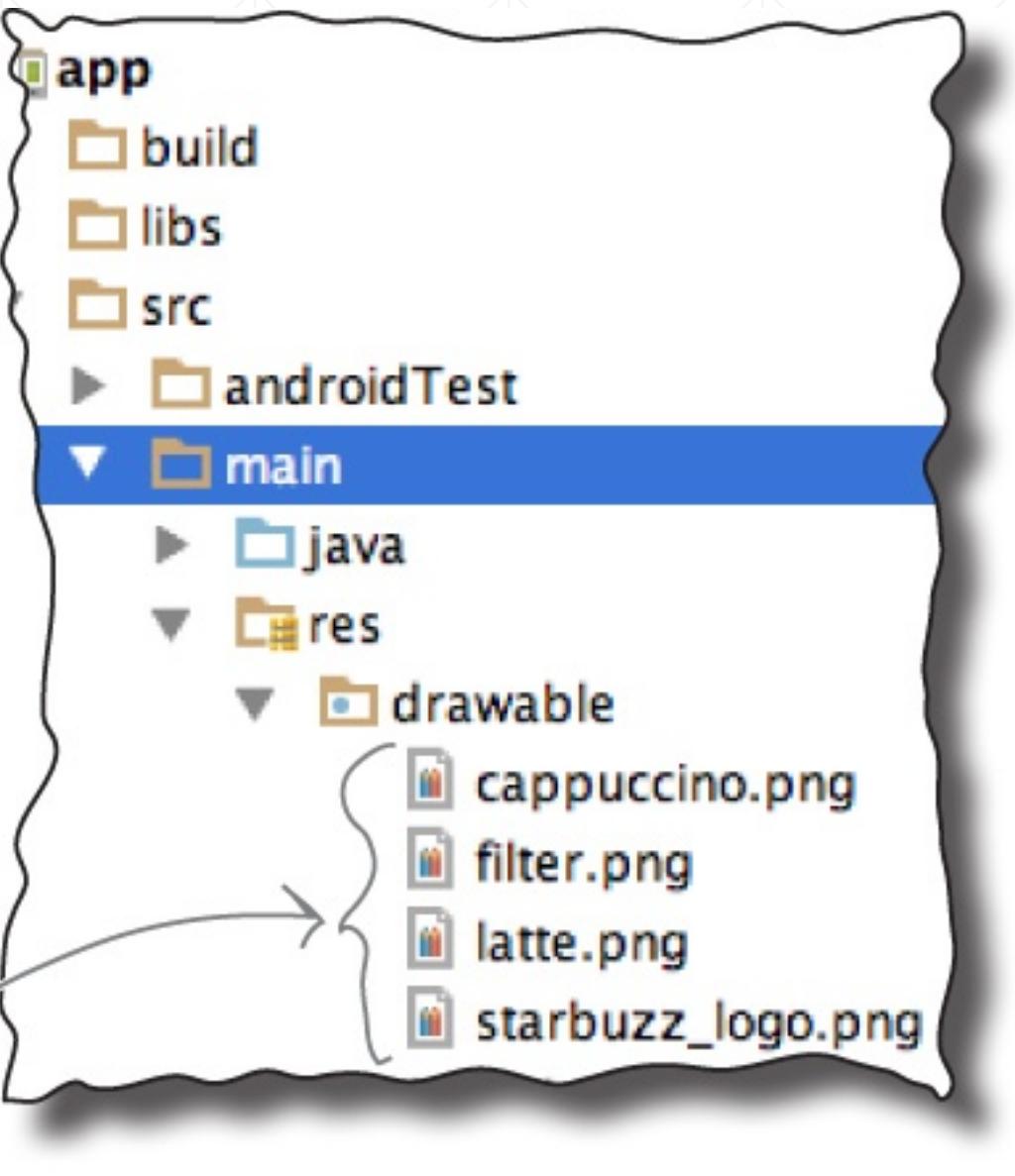
The Drink constructor

These are getters for the private variables.

The String representation of a Drink is its name.



Here are the four image files. You add them to Android Studio by dragging them to the drawable folder.



Images



Drink

name: "Latte"

description: "A couple of espresso shots with steamed milk"

imageResourceId: R.drawable.latte

The image latte.png
is given an ID of
R.drawable.latte.



R.drawable.latte

TopLevelActivity

This is the Starbuzz logo.
We added this image to the →
project on the previous page.

A static list of options →



TopLevelActivity

```
<ImageView
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="100dp"
```

```
    android:src="@drawable/starbuzz_logo"
```

```
    android:contentDescription="@string/starbuzz_logo" />
```

These are the dimensions we want the image to have.

The source of the image is the starbuzz_logo.png

file we added to the app.

Adding a content description makes your app more accessible.

```
<resources>
```

```
    ...
```

```
        <string name="starbuzz_logo">Starbuzz logo</string>
```

```
</resources>
```

TopLevelActivity

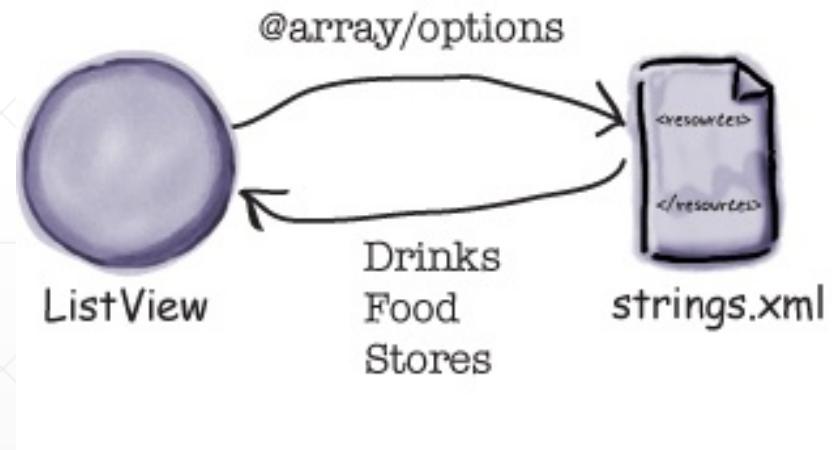
```
<ListView <-- This defines the list view.
```

```
    android:id="@+id/list_options"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:entries="@array/options" />
```

```
<resources>  
    ...  
    <string-array name="options">  
        <item>Drinks</item>  
        <item>Food</item>  
        <item>Stores</item>  
    </string-array>  
</resources>
```

The values in
the list view are
defined by the
options array.

TopLevelActivity



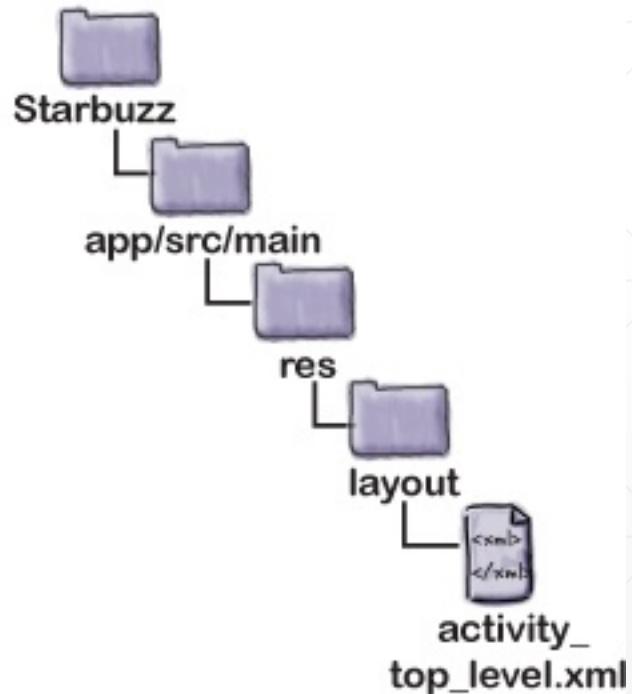
The `entries` attribute populates the `ListView` with values from the `options` array. Each item in the `ListView` is a `TextView`.



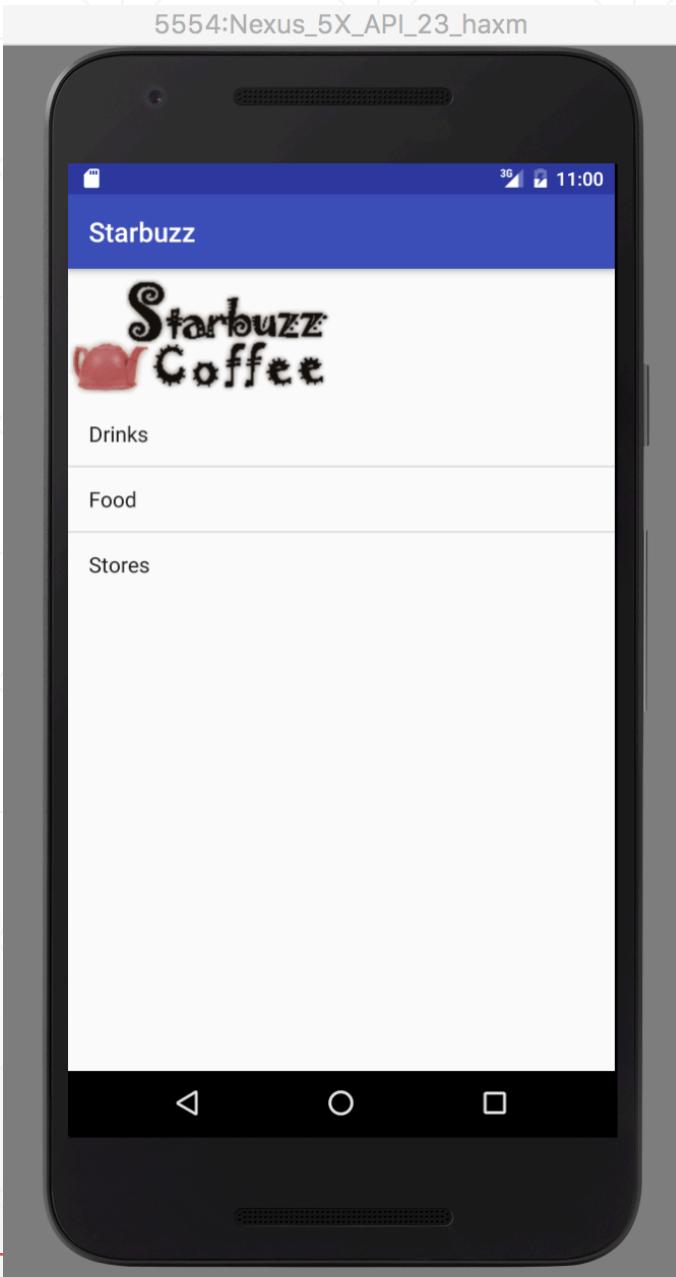
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" ←  
    tools:context=".TopLevelActivity" >
```

We're using a linear layout with a vertical orientation. This will display our list view directly underneath the Starbuzz logo.

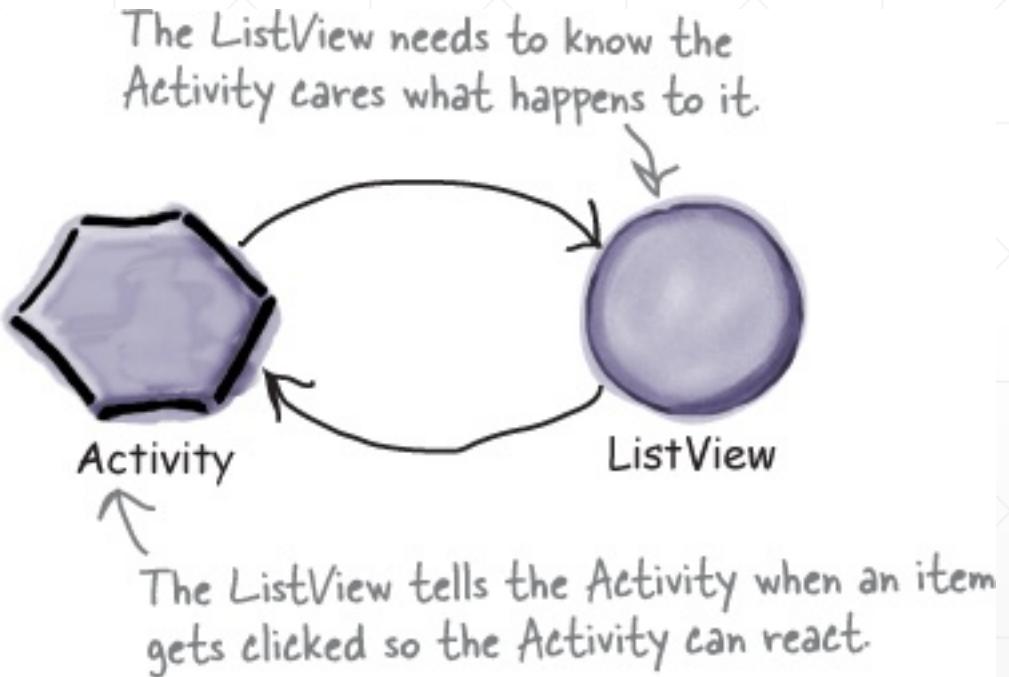
```
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="100dp"  
    android:src="@drawable/starbuzz_logo"  
    android:contentDescription="@string/starbuzz_logo" />  
  
<ListView  
    android:id="@+id/list_options"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:entries="@array/options" />  
  
</LinearLayout>
```



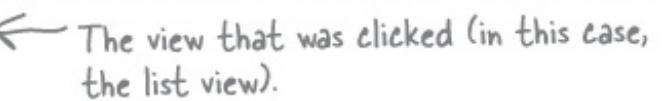
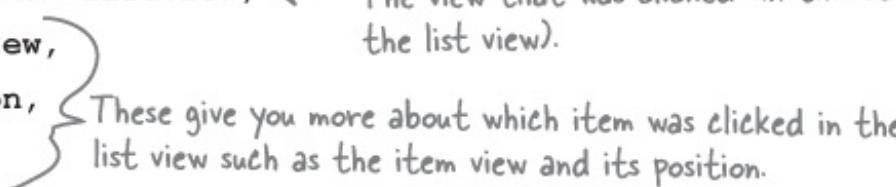
Test Drive



TopLevelActivity



TopLevelActivity

```
AdapterView.OnItemClickListener itemClickListener = new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> listView,  View itemView,  
                           int position,  long id) {  
        if (position == 0) {  
            Intent intent = new Intent(TopLevelActivity.this, DrinkCategoryActivity.class);  
            startActivity(intent);   
        }  
    }  
};
```

OnItemClickListener is a nested class with the AdapterView class. A ListView is a subclass of AdapterView.

Drinks is the first item in the list view, so it's at position 0.

The intent's coming from TopLevelActivity.

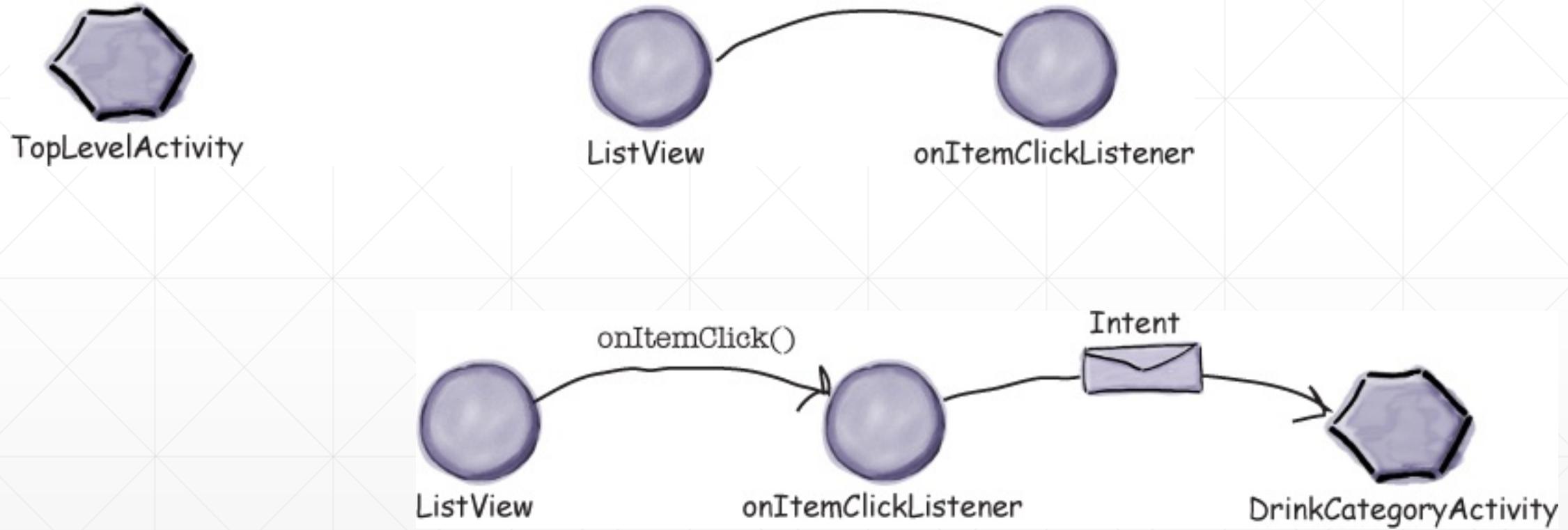
It needs to launch DrinkCategoryActivity.

TopLevelActivity

```
AdapterView.OnItemClickListener itemClickListener = new AdapterView.OnItemClickListener() {  
    public void onItemClick(AdapterView<?> listView,  
        ...  
    }  
};  
  
ListView listView = (ListView) findViewById(R.id.list_options);  
listView.setOnItemClickListener(itemClickListener);
```

 This is the listener we just created.

TopLevelActivity

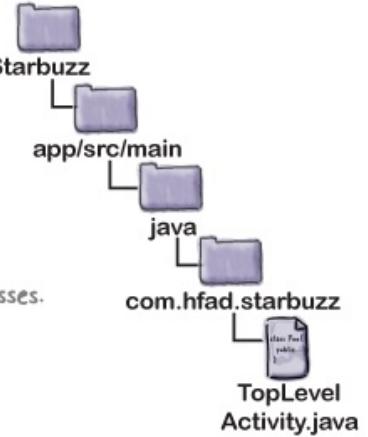


```
package com.hfad.starbuzz;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.ListView; ← We're using these extra classes.
import android.view.View;

public class TopLevelActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_top_level);
        //Create an OnItemClickListener
        AdapterView.OnItemClickListener itemClickListener =
            new AdapterView.OnItemClickListener() {
                public void onItemClick(AdapterView<?> listView,
                    View v,
                    int position,
                    long id) {
                    if (position == 0) {
                        Intent intent = new Intent(TopLevelActivity.this,
                            DrinkCategoryActivity.class);
                        startActivity(intent);
                    }
                }
            };
        //Add the listener to the list view
        ListView listView = (ListView) findViewById(R.id.list_options);
        listView.setOnItemClickListener(itemClickListener); ← Add the listener to the
        }                                         list view.
    }
}
```



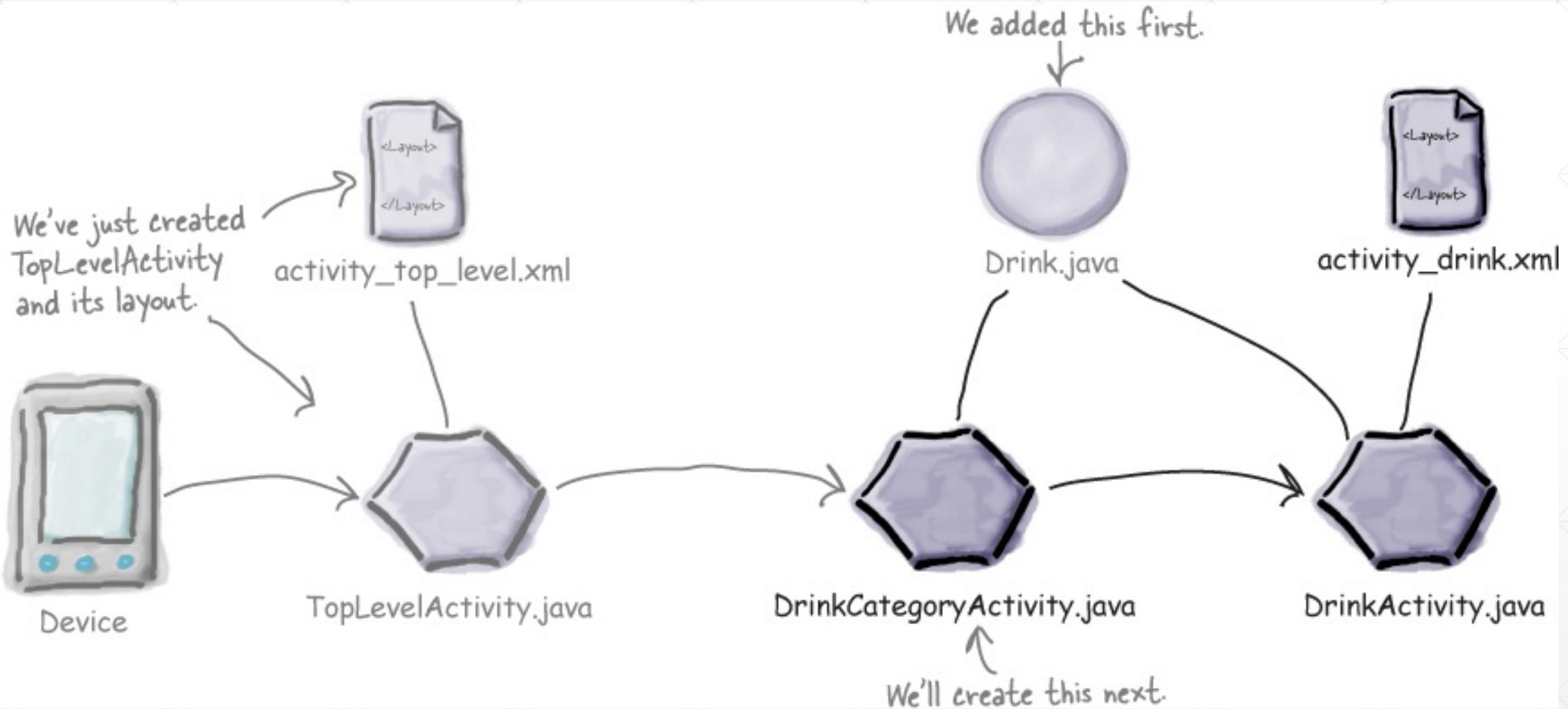
← Create the listener.

↑ Implement its onItemClick() method.

↑ Launch DrinkCategoryActivity if the user clicks on the Drinks item. We'll create this activity next, so don't worry if Android Studio says it doesn't exist.

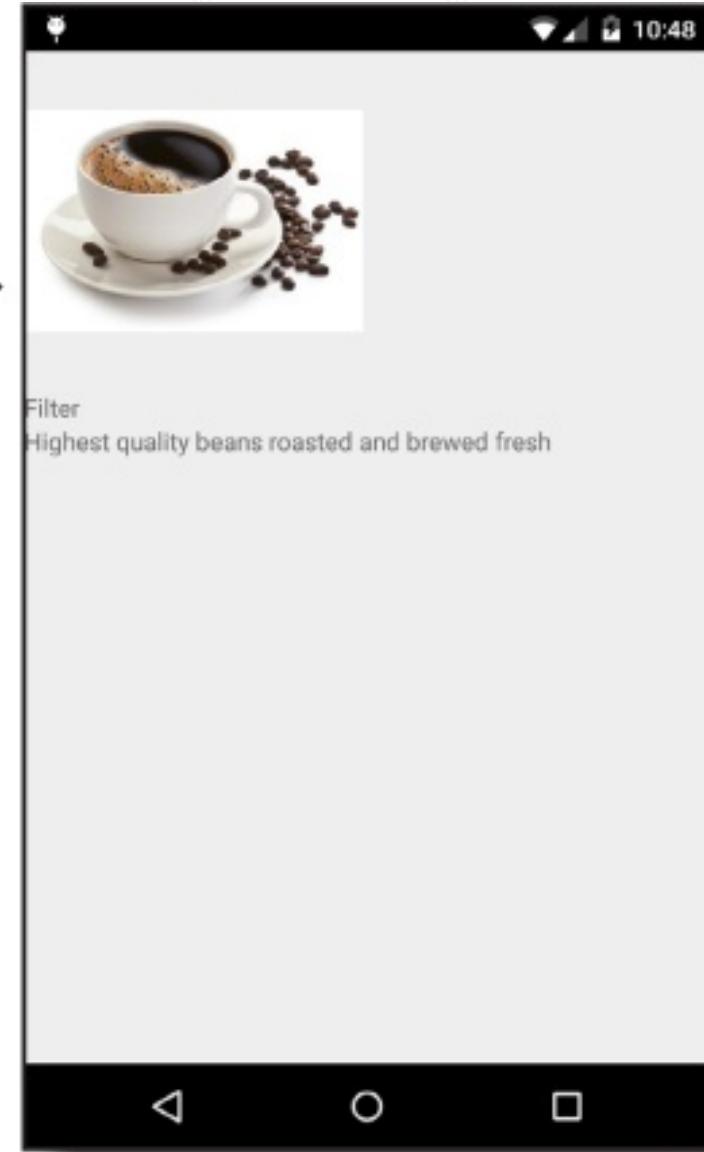
← Add the listener to the list view.

DrinkCategoryActivity





When the user clicks on the Drinks item, activity DrinkCategoryActivity is started.



A list activity comes complete with its own list view so you don't need to add it yourself. You still need to provide it with data, and you'll see how to do that soon.





Configure Activity

Android Studio

Creates a new empty activity

Activity Name:

DrinkCategoryActivity

 Generate Layout File Launcher Activity

Package name:

edu.ltu.starbuzz



If true, a layout file will be generated

Cancel

Previous

Next

Finish

DrinkCategoryActivity

```
package com.hfad.starbuzz;

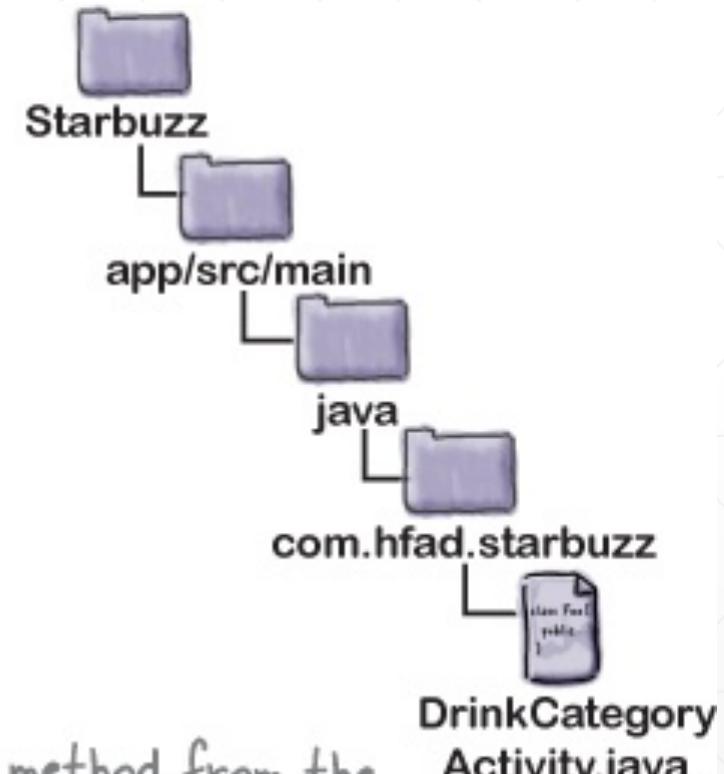
import android.app.ListActivity;
import android.os.Bundle;

public class DrinkCategoryActivity extends ListActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}
```

The activity needs to extend ListActivity, not Activity.

↑

ListActivity inherits the onCreate() method from the Activity class. We'll add code to this method soon.



DrinkCategoryActivity

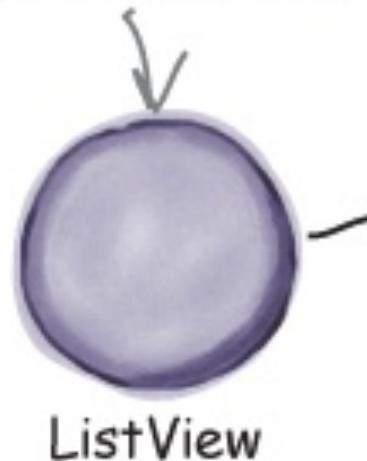
```
<application>
    ...
    <activity
        android:name=".TopLevelActivity" ← Here's the first activity we created.
        android:label="@string/app_name"
        ...
    </activity>
    <activity
        android:name=".DrinkCategoryActivity" ← Here's the new activity.
        android:label="@string/title_activity_drink_category" >
    </activity>
</application>
```

Here's the new activity.
Every activity needs an
entry in `AndroidManifest.xml`.



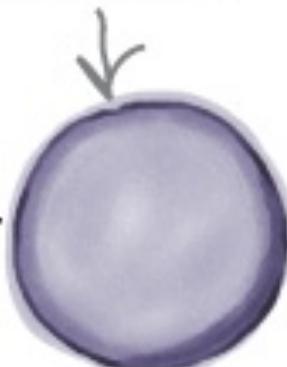
DrinkCategoryActivity

The list view needs
to be populated
with drinks data.



ListView

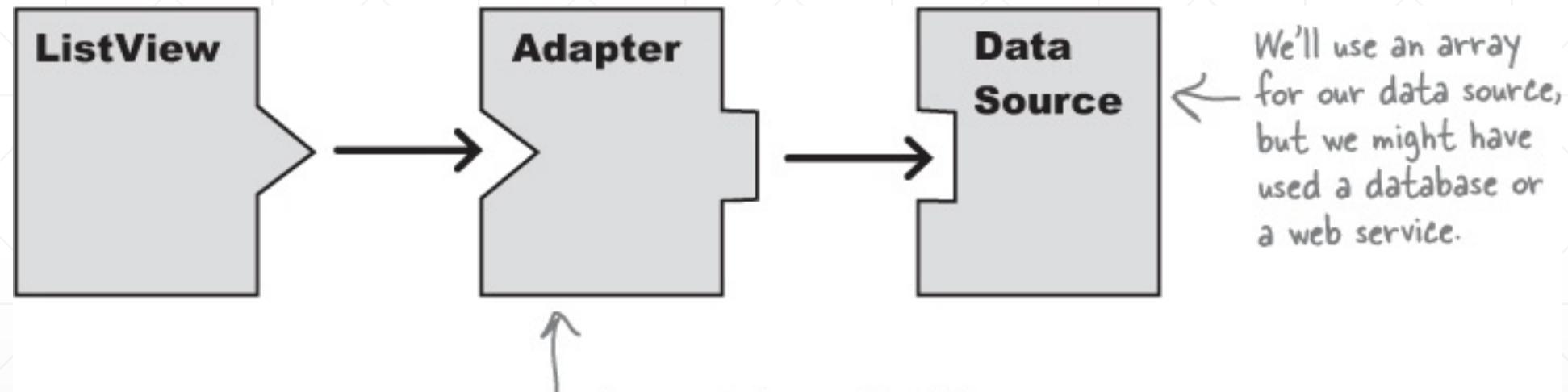
The drink data
needs to come
from the drinks
array in the
Drink class.



Drink.java

drinks

DrinkCategoryActivity



The adapter bridges the gap between the list view and the data source. Adapters allow list views to display data from a variety of sources.

DrinkCategoryActivity

```
ArrayAdapter<Drink> listAdapter = new ArrayAdapter<Drink>(  
    this,  
    android.R.layout.simple_list_item_1,  
    Drink.drinks);
```

this is the current activity.
The Activity class is a
subclass of Context.

→this,
←The array

The array contains Drink objects.

This is a built-in layout
resource. It tells the array
adapter to display each item in
the array in a single text view.

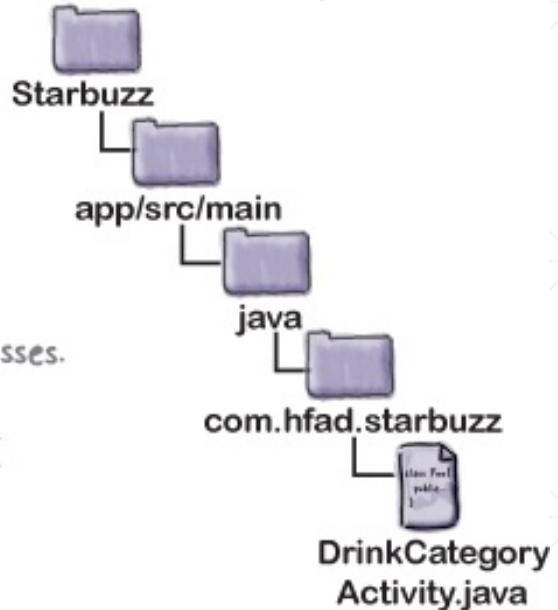
```
ListView listView = getListView();  
listView.setAdapter(listAdapter);
```

```
package com.hfad.starbuzz;

import android.app.ListActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView; ← We're using these extra classes.

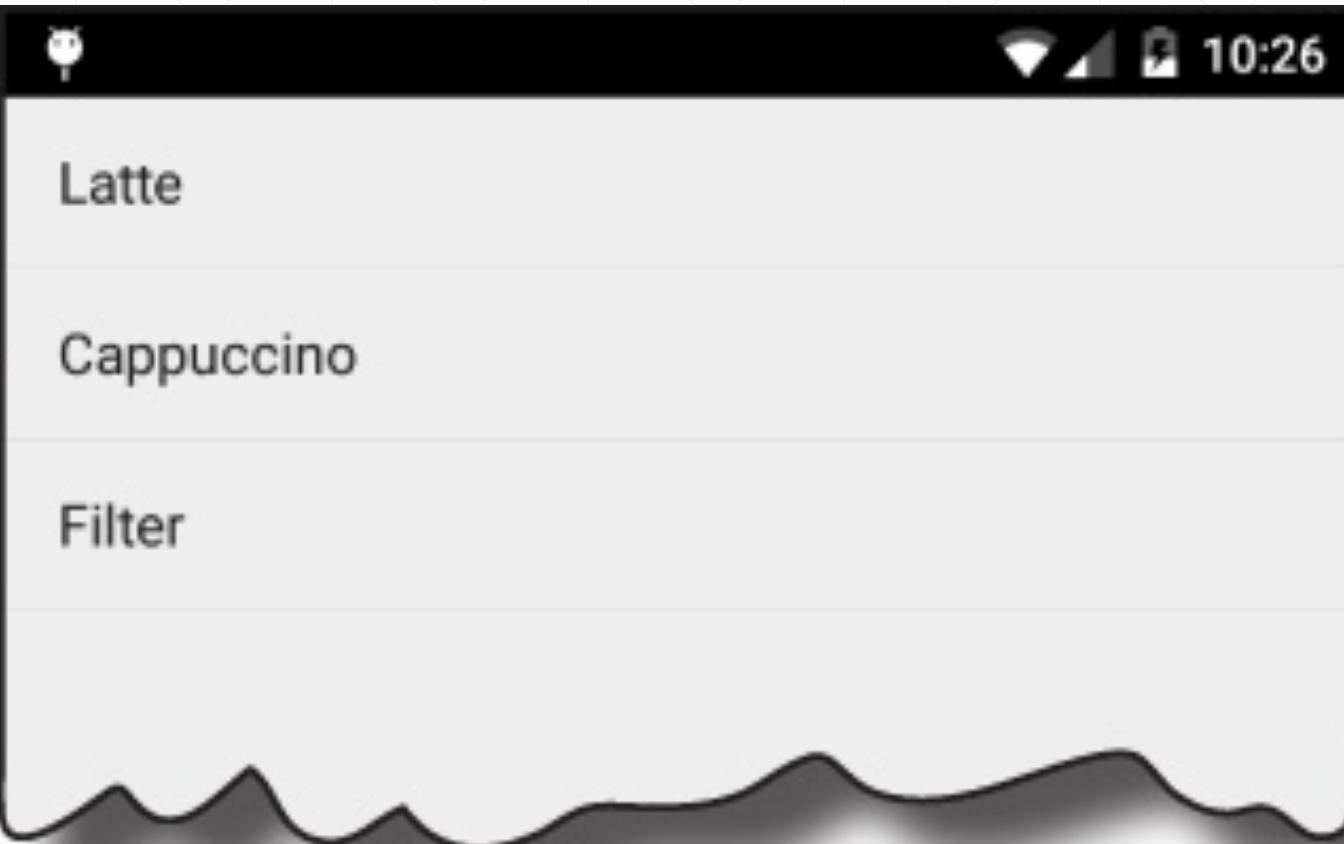
public class DrinkCategoryActivity extends ListActivity {

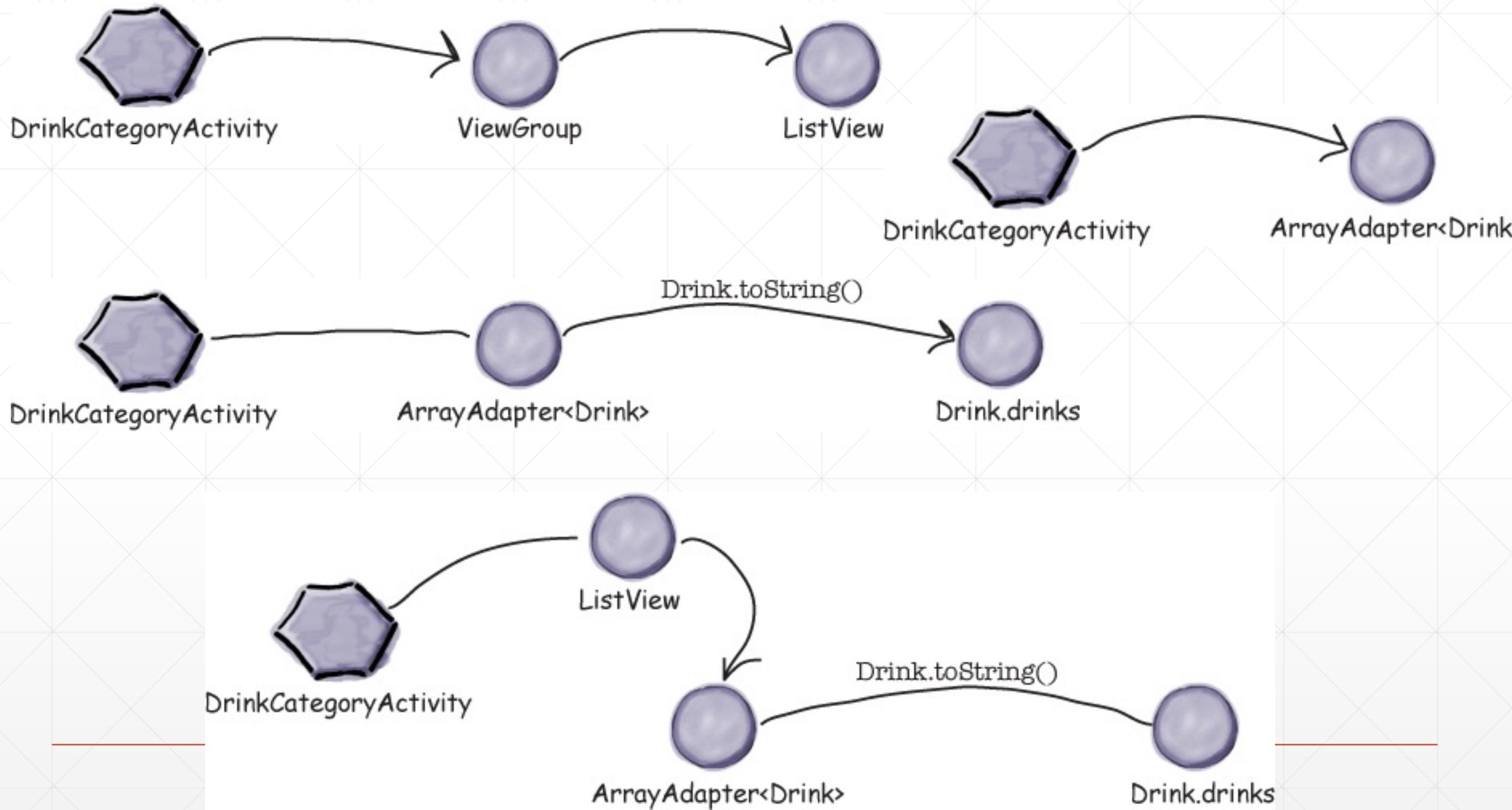
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ListView listDrinks = getListView();
        ArrayAdapter<Drink> listAdapter = new ArrayAdapter<Drink>(
            this,
            android.R.layout.simple_list_item_1,
            Drink.drinks);
        listDrinks.setAdapter(listAdapter);
    }
}
```



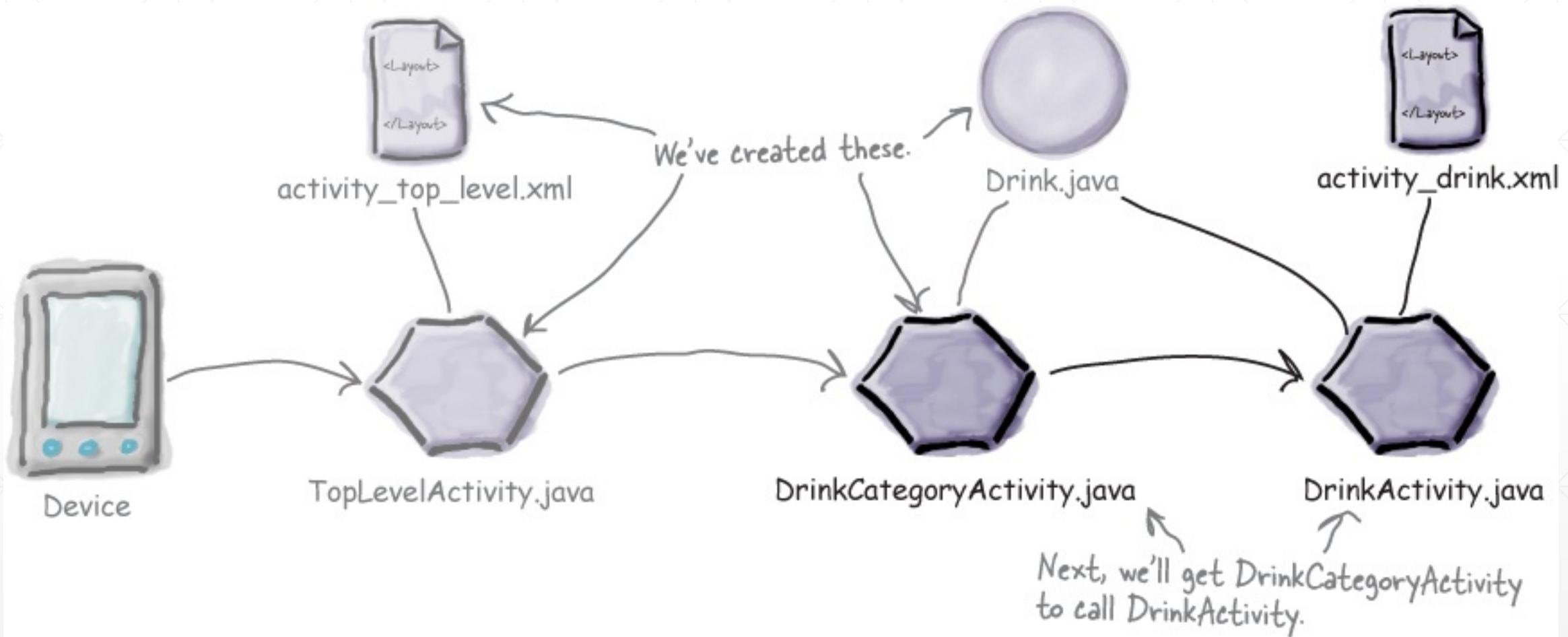
This populates the list view with data from the drinks array.

These are the drinks from
the Drink.drinks array.





DrinkCategoryActivity

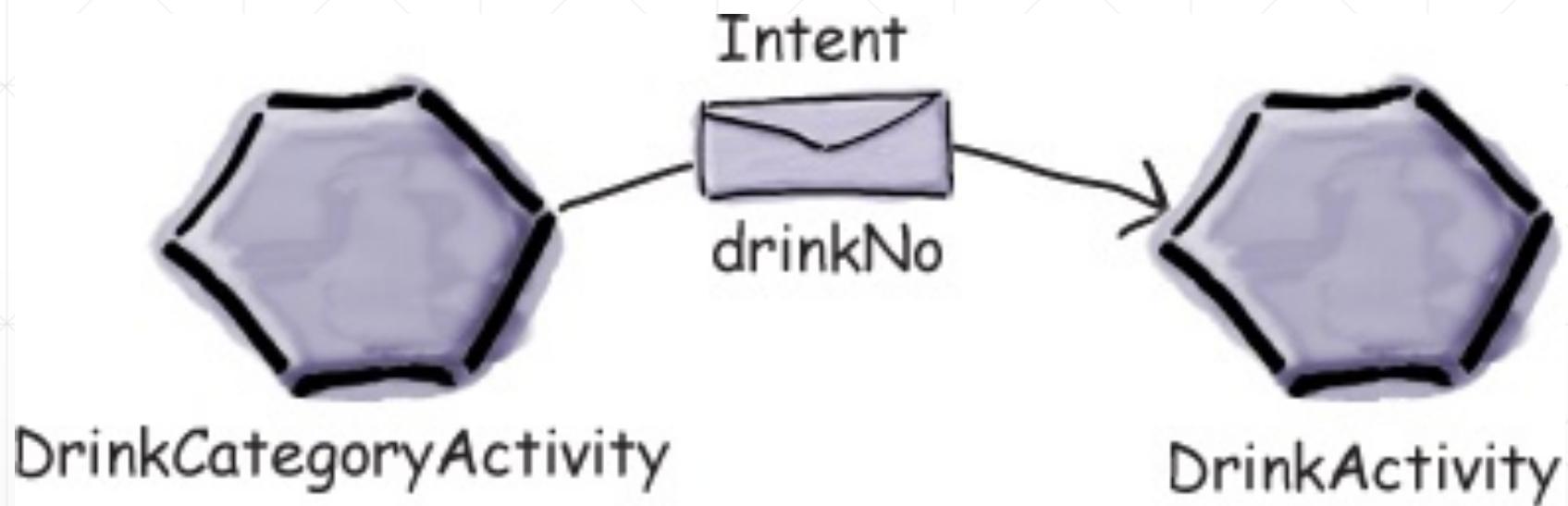


DrinkCategoryActivity

```
public void onListItemClick(ListView listView,  
                           View itemView,  
                           int position,  
                           long id) {  
    //Do something  
}
```

These are the same arguments that the onItemClick() method above has: the list view, the item view that was clicked, its position in the list, and the row ID of the underlying data.

DrinkActivity



DrinkCategoryActivity

```
public void onListItemClick(ListView listView, ← This gets called when an item's clicked.  
                           View itemView,  
                           int position, DrinkCategoryActivity needs to start  
                           long id) { DrinkActivity.  
  
    Intent intent = new Intent(DrinkCategoryActivity.this, DrinkActivity.class);  
    intent.putExtra(DrinkActivity.EXTRA_DRINKNO, (int) id); ← Add the ID of the item that  
    startActivityForResult(intent); was clicked to the intent.  
}  
  
We're using a constant for the name of the  
extra information in the intent so that we know  
DrinkCategoryActivity and DrinkActivity are  
using the same String. We'll add the constant to  
DrinkActivity when we create the activity.
```

```

package com.hfad.starbuzz;

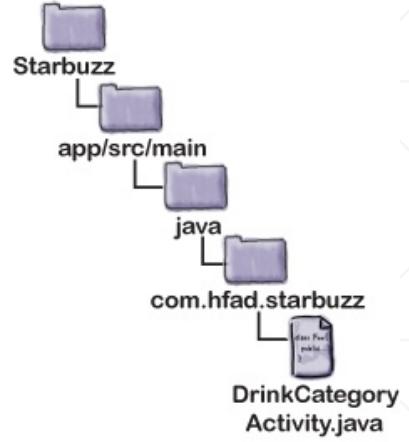
import android.app.ListActivity;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.view.View; ← We're using these extra classes.
import android.content.Intent;

public class DrinkCategoryActivity extends ListActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ListView listDrinks = getListView();
        ArrayAdapter<Drink> listAdapter = new ArrayAdapter<Drink>(
            this,
            android.R.layout.simple_list_item_1,
            Drink.drinks);
        listDrinks.setAdapter(listAdapter);
    }

    @Override
    public void onListItemClick(ListView listView,
                               View itemView,
                               int position,
                               long id) {
        Intent intent = new Intent(DrinkCategoryActivity.this, DrinkActivity.class);
        intent.putExtra(DrinkActivity.EXTRA_DRINKNO, (int) id);
        startActivity(intent);
    }
}

```



Implement the `onListItemClick()` method so that `DrinkActivity` is launched when the user clicks on an item in the list view.

We're adding `DrinkActivity` next, so don't worry if Android Studio says it doesn't exist.



Configure Activity

Android Studio

Creates a new empty activity

Activity Name:

DrinkActivity|

 Generate Layout File

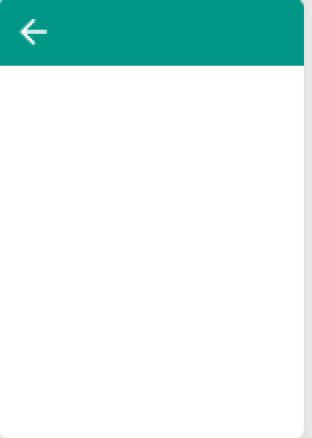
Layout Name:

activity_drink

 Launcher Activity

Package name:

edu.ltu.starbuzz



The name of the activity class to create

Cancel

Previous

Next

Finish

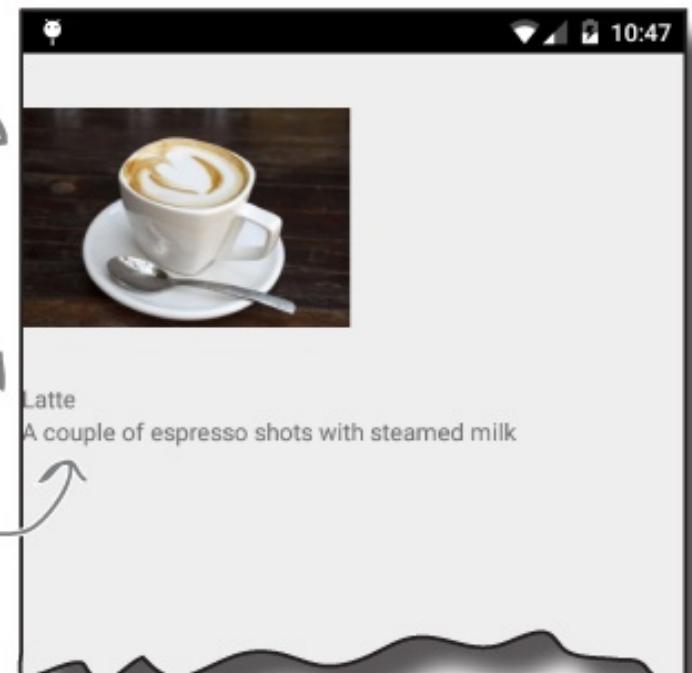
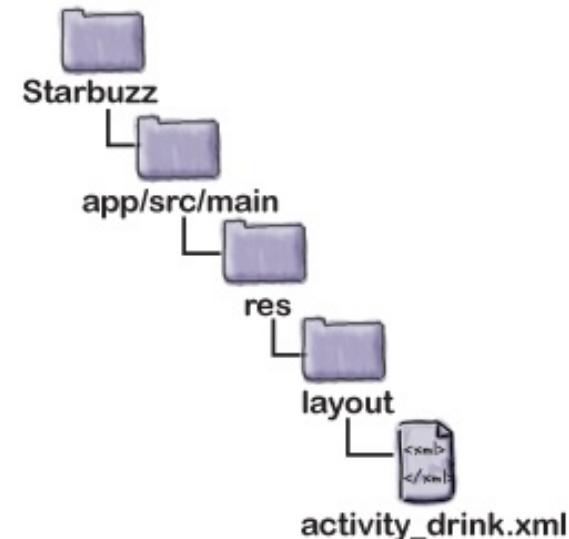
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical"  
    tools:context="com.hfad.starbuzz.DrinkActivity" >
```

```
    <ImageView  
        android:id="@+id/photo"  
        android:layout_width="190dp"  
        android:layout_height="190dp" />
```

```
    <TextView  
        android:id="@+id/name"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content" />
```

```
    <TextView  
        android:id="@+id/description"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content" />
```

```
</LinearLayout>
```



DrinkActivity

```
//Get the drink from the intent  
int drinkNo = (Integer) getIntent().getExtras().get(EXTRA_DRINKNO);  
Drink drink = Drink.drinks[drinkNo];
```

```
package com.hfad.starbuzz;

import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.TextView;

public class DrinkActivity extends Activity {

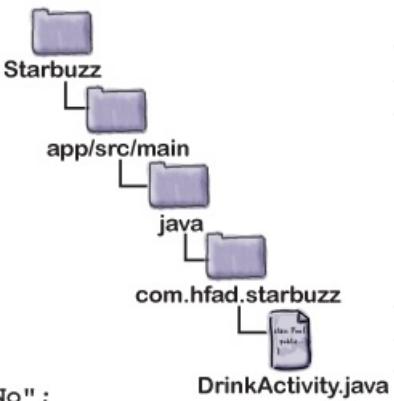
    public static final String EXTRA_DRINKNO = "drinkNo";
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_drink);

        //Get the drink from the intent
        int drinkNo = (Integer) getIntent().getExtras().get(EXTRA_DRINKNO);
        Drink drink = Drink.drinks[drinkNo]; ← Use the drinkNo to get details
                                                of the drink the user chose.

        //Populate the drink image
        ImageView photo = (ImageView) findViewById(R.id.photo);
        photo.setImageResource(drink.getImageResourceId());
        photo.setContentDescription(drink.getName()); ← Populate the views
                                                    with the drink data.

        //Populate the drink name
        TextView name = (TextView) findViewById(R.id.name);
        name.setText(drink.getName()); ← Populate the views
                                    with the drink data.

        //Populate the drink description
        TextView description = (TextView) findViewById(R.id.description);
        description.setText(drink.getDescription());
    }
}
```

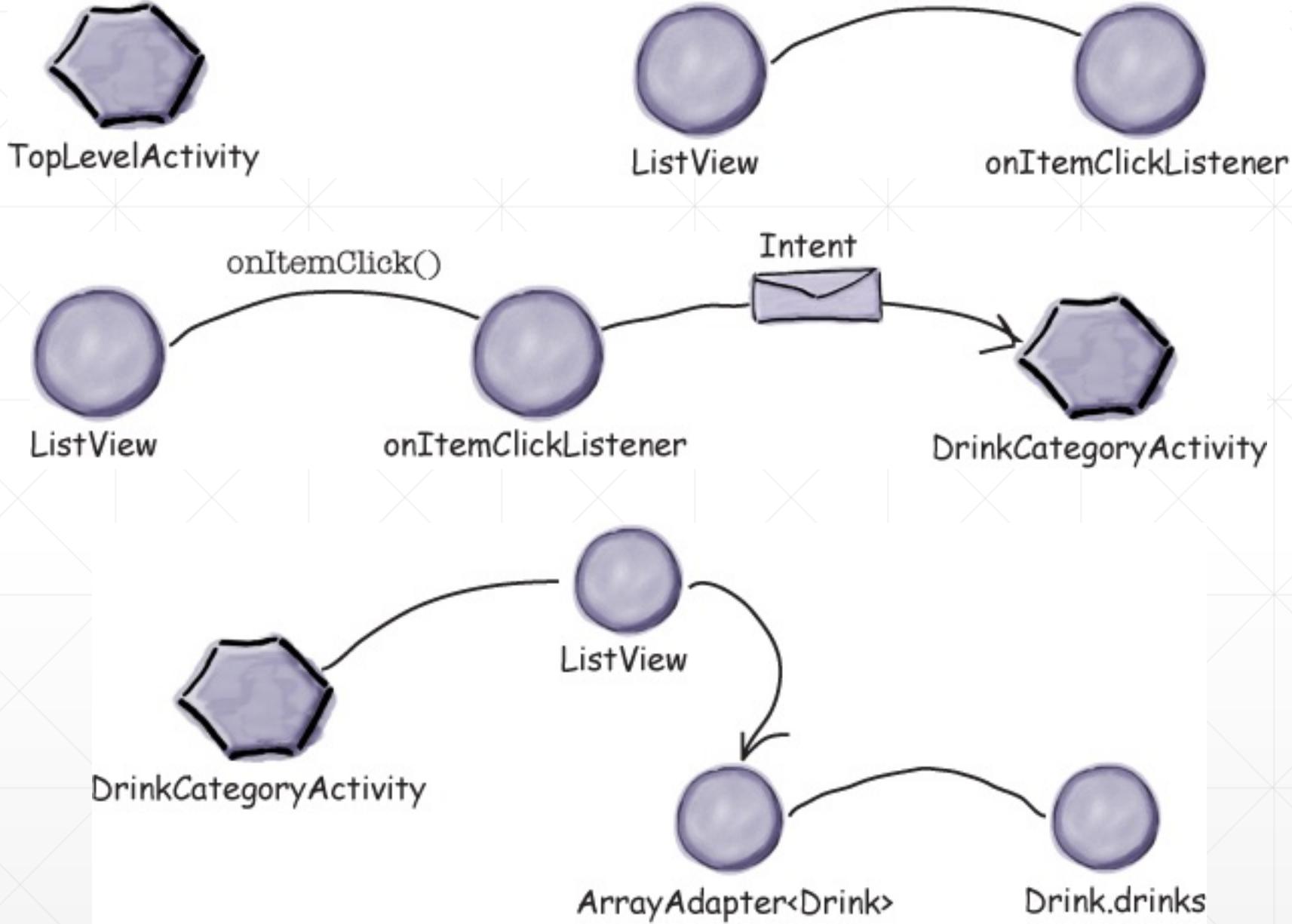


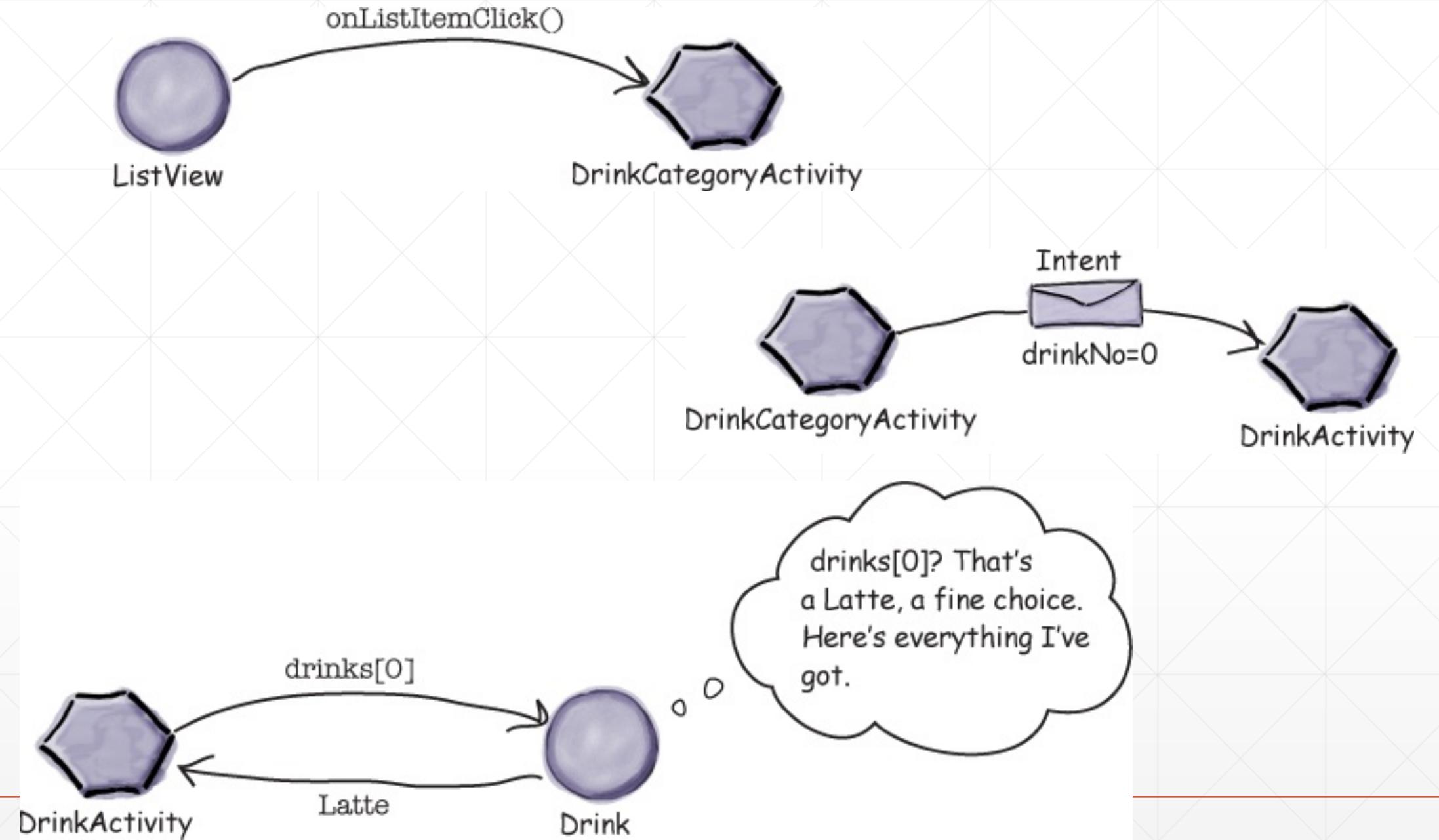
Add EXTRA_DRINKNO as a constant.

← Use the drinkNo to get details
of the drink the user chose.

← Populate the views
with the drink data.

← Populate the views
with the drink data.





Bullet Points

- Sort your ideas for activities into top-level activities, category activities, and detail/edit activities. Use the category activities to navigate from the top-level activities to the detail/edit activities.
 - Image resources go in one or more of the drawable* folders. You reference them in your layout using @drawable/image_name. You access them in your activity code using R.drawable.image_name.
 - An ImageView holds an image. Add it to your layout using <ImageView>. Use android:src to set its source, and android:contentDescription to give it an accessible label. The equivalent methods in Java are setImageResource() and setContentDescription().
 - A ListView displays items in a list. Add it to your layout using <ListView>.
 - Use android:entries in your layout to populate the items in your list views from an array defined in strings.xml.
-

Bullet Points

- A ListActivity is an Activity that comes with a ListView. You get a reference to the ListView using getListView().
 - A ListActivity has its own default layout, but you can replace it with your own.
 - An adapter acts as a bridge between an AdapterView and a data source. ListViews and Spinners are both types of AdapterView.
 - An ArrayAdapter is an adapter that works with arrays.
 - Handle click events on Buttons using android:onClick in the layout code.
 - Handle click events on a ListView in a ListActivity by implementing the onListItemClick() method.
 - Handle click events elsewhere by creating a listener and implementing its click event.
-