TASK GUIDE (B4.06)

A. **Objectives.**

Student will start to write the code for MainActivity which show data in RecyclerView and make them can switch dinamically.

B. **Requirements.**

Hardware:

- 2 GB RAM minimum, 8 GB RAM recommended
- 2 GB of available disk space minimum, 4 GB Recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality

Software:

- Microsoft Windows 7/8/10 (32-bit or 64-bit)
- JDK 8
- Android Studio IDE (Minimum 3.2) with AndroidX library.

C. **Resources.**

Documents:

- Guide

Supplement files:

Test code:

- TestB4MultimediaResources061.java

D. **Task Description.**

Student start to write code in MainActivity.java.

E. Specification.

1. To load data in RecyclerView, first create a new Java class for Data item with name ‘DataItem.java’, and modify the code, like below :

```
package org.aplas.animaltour;

public class DataItem {

    private String title;
    private String info;
    private int color;
    private int icon;

    public DataItem(String mTitle, String mInfo, int mColor, int mIcon) {
        title = mTitle;
        info = mInfo;
        color = mColor;
        icon = mIcon;
    }

    public String getTitle() {
        return title;
    }
    public String getInfo() {
        return info;
    }
    public int getColor() {
        return color;
    }
    public int getIcon() {
        return icon;
    }
    public void setTitle(String val) {
        this.title =val;
    }
    public void setInfo(String val) {
        this.info = val;
    }
    public void setColor(int val) {
        this.color = val;
    }
    public void setIcon(int val) {
        this.icon = val;
    }
}
```

A data contains 4 fields namely:

- **Title**: a string to represent an animal name from “animal_titles” string resource.
- **Info**: a string to represent an animal description from “animal_info” string resource.
- **Color**: a number to represent a color from “animal_color” color resource.
- **Icon**: a number to represent an animal icon resource id from “animal_icon” string resource.

2. Then, create again a new Java class for movement of RecyclerView items with name ‘ItemMoveCallback.java’. Change the code, like below :

```
package org.aplas.animaltour;

import androidx.annotation.NonNull;
import androidx.recyclerview.widget.ItemTouchHelper;
import androidx.recyclerview.widget.RecyclerView;

public class ItemMoveCallback extends ItemTouchHelper.Callback {
    private final ItemTouchHelperContract mAdapter;
    public ItemMoveCallback(ItemTouchHelperContract adapter) {
        mAdapter = adapter;
    }

    @Override
    public boolean isLongPressDragEnabled() { return true; }

    @Override
    public boolean isItemViewSwipeEnabled() { return false; }

    @Override
    public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int i) {  }

    @Override
    public int getMovementFlags(RecyclerView recyclerView,
        RecyclerView.ViewHolder viewHolder) {
        int dragFlags = ItemTouchHelper.UP | ItemTouchHelper.DOWN;
        return makeMovementFlags(dragFlags, 0);
    }

    @Override
    public boolean onMove(RecyclerView recyclerView,
        RecyclerView.ViewHolder viewHolder, RecyclerView.ViewHolder target) {
        mAdapter.onRowMoved(viewHolder.getAdapterPosition(), target.getAdapterPosition());
        return true;
    }

    @Override
    public void onSelectedChanged(RecyclerView.ViewHolder viewHolder,int actionState) {
        if (actionState != ItemTouchHelper.ACTION_STATE_IDLE) {
          if (viewHolder instanceof DataAdapter.ViewHolder) {
            DataAdapter.ViewHolder myViewHolder=(DataAdapter.ViewHolder) viewHolder;
            mAdapter.onRowSelected(myViewHolder);
          }
        }
        super.onSelectedChanged(viewHolder, actionState);
    }

    @Override
    public void clearView(RecyclerView recyclerView, RecyclerView.ViewHolder viewHolder) {
        super.clearView(recyclerView, viewHolder);
        if (viewHolder instanceof DataAdapter.ViewHolder) {
            DataAdapter.ViewHolder myViewHolder =(DataAdapter.ViewHolder) viewHolder;
            mAdapter.onRowClear(myViewHolder);
        }
    }

    public interface ItemTouchHelperContract {
        void onBindViewHolder(DataAdapter.ViewHolder holder, int position);
        void onRowMoved(int fromPosition, int toPosition);
        void onRowSelected(DataAdapter.ViewHolder myViewHolder);
        void onRowClear(DataAdapter.ViewHolder myViewHolder);
    }
}
```

3. Then, create again a new Java class for Data adapter with name ‘DataAdapter.java’. Then modify the code, like below :

```
package org.aplas.animaltour;

import …

public class DataAdapter extends RecyclerView.Adapter<DataAdapter.ViewHolder>
       implements ItemMoveCallback.ItemTouchHelperContract {


}
```

4. In the ‘DataAdapter’ class, create some variables and a constructor, like below:

```
private ArrayList<DataItem> itemList;
private Context mContext;
private OnItemClickListener mListener;

// Constructor of the class
public DataAdapter(Context context, ArrayList<DataItem> itemList) {
    mContext = context;
    this.itemList = itemList;
}
```

Then, add an Override method to specify the layout of each row, like below:

```
@Override
public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
   View v = LayoutInflater.from(mContext).
       inflate(R.layout.layout_data, parent, false);
   return new ViewHolder(v,mListener);
}
```

Then, add an Override method to load data for each row elements, like below:

```
@Override
public void onBindViewHolder(final ViewHolder holder, int position) {
   DataItem currData = itemList.get(position);
   holder.bindTo(currData);
}
```

To provide data item movement in RecyclerView, put this code:

```
@Override
public void onRowMoved(int fromPosition, int toPosition) {
   if (fromPosition < toPosition) {
       for (int i = fromPosition; i < toPosition; i++) {
          Collections.swap(itemList, i, i + 1);
       }
   } else {
       for (int i = fromPosition; i > toPosition; i--) {
          Collections.swap(itemList, i, i - 1);
       }
   }
   notifyItemMoved(fromPosition, toPosition);
}

    @Override
    public void onRowSelected(ViewHolder myViewHolder) {
        myViewHolder.rowView.setBackgroundColor(Color.GRAY);

    }

    @Override
    public void onRowClear(ViewHolder myViewHolder) {
        myViewHolder.rowView.setBackgroundColor(Color.WHITE);

    }
```

Then, add an Override method to get list size, like below:

```
@Override
public int getItemCount() {
   return itemList.size();
}
```

Add an interface and a method to handle onClickListener, like below:

```
    public interface OnItemClickListener {
        void onItemClick(View view, int position);
    }

    public void setOnItemClickListener(OnItemClickListener listener) {
        mListener = listener;
    }
```

5. Still in the ‘DataAdapter’ class, add a static class ‘ViewHolder’ to initiate views of rows, like below:

```
    public static class ViewHolder extends RecyclerView.ViewHolder {


    }
```

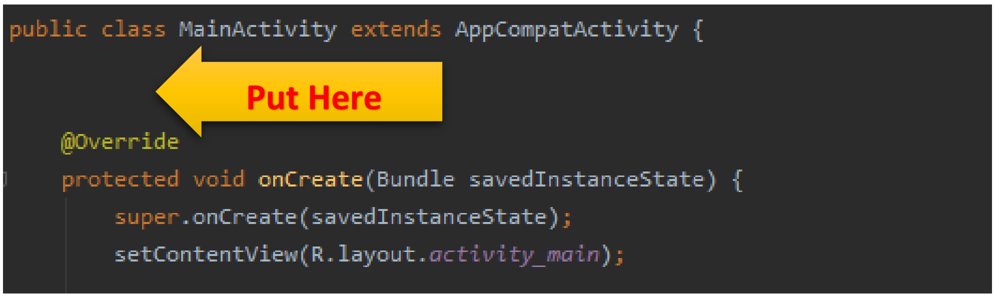
In the 'ViewHolder' class, create some variables and a constructor, like below:

```
public TextView title;
public TextView info;
public ImageView icon;
public View rowView;

public ViewHolder(View itemView, final OnItemClickListener listener) {
   super(itemView);
   title = (TextView) itemView.findViewById(R.id.animalTitle);
   info = (TextView) itemView.findViewById(R.id.animalInfo);
   icon = (ImageView) itemView.findViewById(R.id.animalIcon);
   rowView = itemView;

   itemView.setOnClickListener(new View.OnClickListener() {
       @Override
       public void onClick(View v) {
           if (listener != null) {
               int position = getAdapterPosition();
               if (position != RecyclerView.NO_POSITION) {
                  listener.onItemClick(v,position);
               }
           }
       }
   });
}
```

In the 'ViewHolder' class, add a method 'bindTo' to set the elements in RecyclerView item regarding to current data, like below:

```
public void bindTo(DataItem currData){
   title.setText(currData.getTitle());
   title.setBackgroundColor(currData.getColor());
   info.setText(currData.getInfo());
   icon.setBackgroundResource(currData.getIcon());
   AnimationDrawable anim = (AnimationDrawable) icon.getBackground();
   anim.start();
}
```

6. Open "MainActivity.java" in java folder.
7. Declare all variables that represents all widgets and global variable in activity_main.xml.



```
public class MainActivity extends AppCompatActivity {


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
```

Using this template:

```
private <Type> <variable_name>;
```

All the variables are:

Data Type	Variable_name	Group
RecyclerView	rcView	**Widget**
DataAdapter	mAdapter	Android class

8. In the onCreate method, define all **Widgets** variables, which has been declared in point 1, to the related widget id using this template:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);

   <variable_name> = (<Widget_type>)findViewById(R.id.<widget_id>);
   .
   .
}
```

Then set the RecyclerView layout manager to Linier, like below:

```
rcView.setLayoutManager(new LinearLayoutManager(this));
```

9. The, create a new function 'getId' to get a project resource id, like below.

```
public static int getId(String resourceName, Class<?> c) {
   try {
      Field idField = c.getDeclaredField(resourceName);
      return idField.getInt(idField);
   } catch (Exception e) {
      return -1;
   }
}
```

10. Create a new function 'getAnimalData' to get animal data from String and Color resource also invertebrate data, then submit them to a ArrayList<DataItem>, like below.

```
private ArrayList<DataItem> getAnimalData() {
   //Get the resources from the XML file
   String[] listTitles = getResources().getStringArray(R.array.animal_titles);
   int[] listColor = getResources().getIntArray(R.array.animal_color);
   String[] listInfo = getResources().getStringArray(R.array.animal_info);
   String[] listIcon = getResources().getStringArray(R.array.animal_icon);

   ArrayList<DataItem> data = new ArrayList<>();

   //Create the ArrayList with the titles and information
   for (int i=0; i<listTitles.length; i++) {
       data.add(new DataItem(listTitles[i],listInfo[i],listColor[i],
          getId(listIcon[i],R.drawable.class)));
   }
   //Add Invertebrates
   data.add(new DataItem("Invertebrates",this.getString(R.string.invert_info),
       this.getColor(R.color.invert_color), R.drawable.animinverts));

   return data;
}
```

11. Create a new method to load the animal data of RecyclerView, like below.

```
private void loadAnimalData() {
    //Initialize the adapter and set it ot the RecyclerView
    mAdapter = new DataAdapter(this, getAnimalData());

    ItemTouchHelper.Callback callback = new ItemMoveCallback(mAdapter);
    ItemTouchHelper touchHelper = new ItemTouchHelper(callback);
    touchHelper.attachToRecyclerView(rcView);

    rcView.setAdapter(mAdapter);

    //Notify the adapter of the change
    mAdapter.notifyDataSetChanged();
}
```

12. Back to ‘onCreate’ method, call the ‘loadAnimalData’ here.

13. We will use this scenario:
    - When an animal clicked, the second activity (MediaActivity) will be opened and show the media (local video and YouTube video) of clicked animal.
    - When invertebrate animal clicked, the thirc activity (InvertActivity) will be opened.

    First, we have to make a method ‘openMediaActivity’ with a string parameter (animal name), to open MediaActivity as intent with inserting ‘TITLE_ANIMAL’ as extra and using transition animation ‘slide_in_left’ and ‘slide_out_left’ when change the active activity with ‘overridePendingTransition’. The code like below:

```
private void openMediaActivity(String title) {
   Intent media = new Intent(getApplicationContext(),MediaActivity.class);
   media.putExtra("TITLE_ANIMAL", title);
   startActivity(media);
   overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_left);
}
```

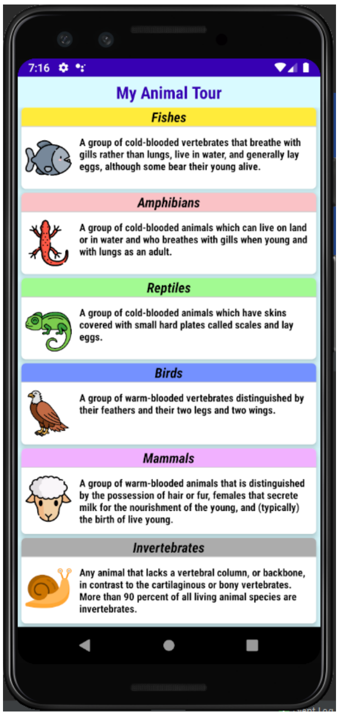
Then, we have to make a method ‘openInvertActivity’ to open InvertActivity as intent using transition animation ‘fade_in’ and ‘fade_out’ when change the active activity with ‘overridePendingTransition’. The code like below:

```
private void openInvertActivity() {
   Intent intent = new Intent(getApplicationContext(),InvertActivity.class);
   startActivity(intent);
   overridePendingTransition(R.anim.fade_in, R.anim.fade_out);
}
```

In onCreate method, we have to create an item click listener for the RecyclerView. When invertebrates is clicked, it will call ‘openInvertActivity’ method, otherwise will call ‘openMediaActivity’ with related item as input parameter. Set the item click listener for mAdapter like below:

```
mAdapter.setOnItemClickListener(new DataAdapter.OnItemClickListener() {
   @Override
   public void onItemClick(View view, int position) {
      TextView itemTitle = (TextView) view.findViewById(R.id.animalTitle);
      if (itemTitle.getText().toString().equals("Invertebrates")) {
         openInvertActivity();
      } else {
         openMediaActivity(itemTitle.getText().toString());
      }
   }
});
```

14. Try to run the app and make sure all animals (6 animals) are shown on MainActivity like below:



7:16

My Animal Tour

Fishes

A group of cold-blooded vertebrates that breathe with gills rather than lungs, live in water, and generally lay eggs, although some bear their young alive.

Amphibians

A group of cold-blooded animals which can live on land or in water and who breathes with gills when young and with lungs as an adult.

Reptiles

A group of cold-blooded animals which have skins covered with small hard plates called scales and lay eggs.

Birds

A group of warm-blooded vertebrates distinguished by their feathers and their two legs and two wings.

Mammals

A group of warm-blooded animals that is distinguished by the possession of hair or fur, females that secrete milk for the nourishment of the young, and (typically) the birth of live young.

Invertebrates

Any animal that lacks a vertebral column, or backbone, in contrast to the cartilaginous or bony vertebrates. More than 90 percent of all living animal species are invertebrates.

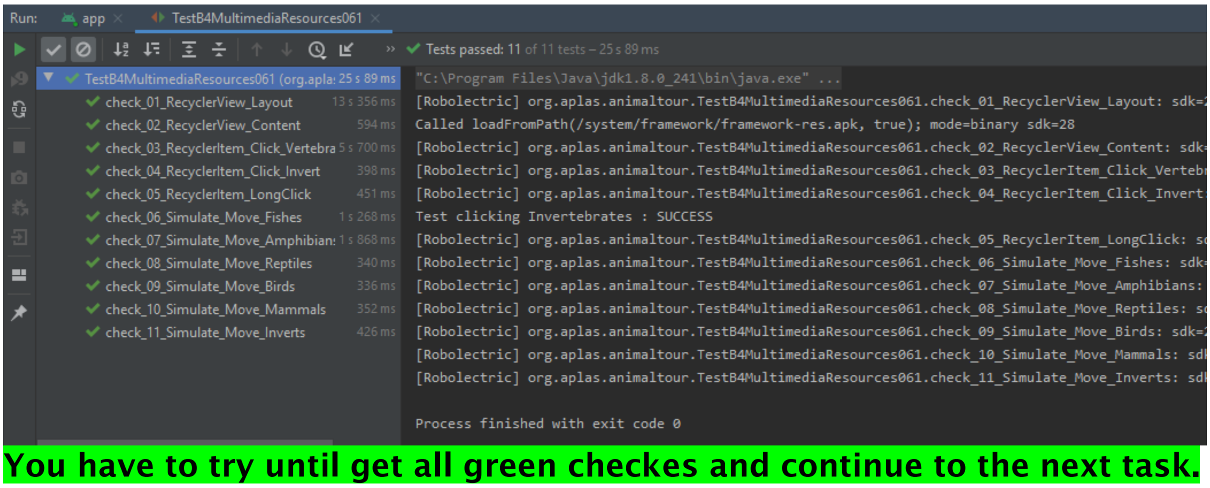
If not yet, you must modify you file ‘**layout_data.xml**’ to provide enough space for 6 animals shown on single screen.

F. Testing.

1. Copy “TestB4MultimediaResources061.java” file to “org.aplas.animaltour (test)” folder.

2. Right click on the “TestB4MultimediaResources061.java” file then choose Run. It may take long time to execute.

3. Get the result of your task. If passed you will get green check like below. If the test failed, you will get orange check get the messages and you must check your work again.



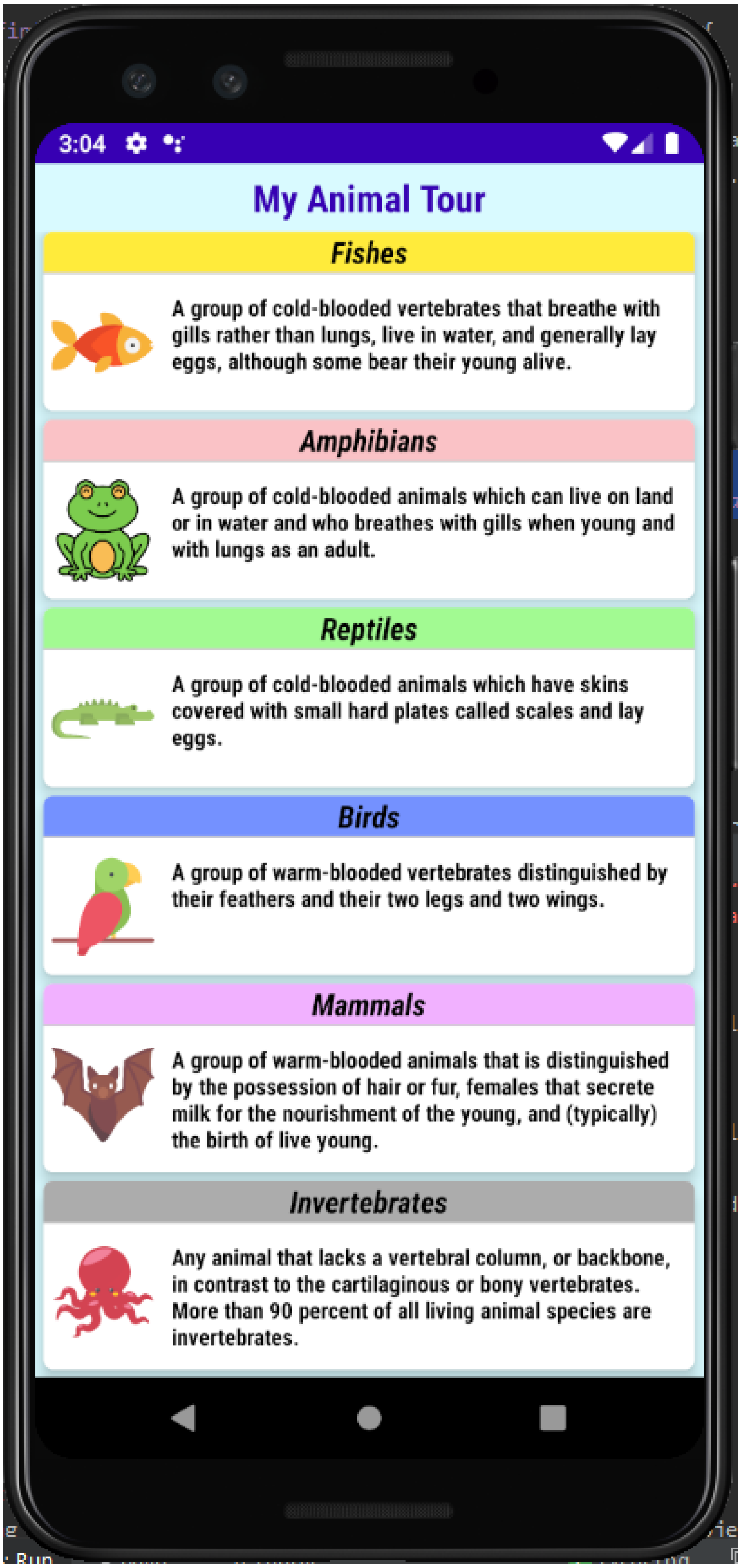
✓ Tests passed: 11 of 11 tests – 25 s 89 ms

```
▼ ✓ TestB4MultimediaResources061 (org.apla: 25 s 89 ms
    ✓ check_01_RecyclerView_Layout          13 s 356 ms
    ✓ check_02_RecyclerView_Content             594 ms
    ✓ check_03_RecyclerItem_Click_Vertebra 5 s 700 ms
    ✓ check_04_RecyclerItem_Click_Invert        398 ms
    ✓ check_05_RecyclerItem_LongClick           451 ms
    ✓ check_06_Simulate_Move_Fishes        1 s 268 ms
    ✓ check_07_Simulate_Move_Amphibian:    1 s 868 ms
    ✓ check_08_Simulate_Move_Reptiles           340 ms
    ✓ check_09_Simulate_Move_Birds              336 ms
    ✓ check_10_Simulate_Move_Mammals            352 ms
    ✓ check_11_Simulate_Move_Inverts            426 ms
```

```
"C:\Program Files\Java\jdk1.8.0_241\bin\java.exe" ...
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_01_RecyclerView_Layout: sdk=
Called loadFromPath(/system/framework/framework-res.apk, true); mode=binary sdk=28
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_02_RecyclerView_Content: sdk
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_03_RecyclerItem_Click_Vertebr
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_04_RecyclerItem_Click_Invert
Test clicking Invertebrates : SUCCESS
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_05_RecyclerItem_LongClick: s
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_06_Simulate_Move_Fishes: sdk
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_07_Simulate_Move_Amphibians:
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_08_Simulate_Move_Reptiles: s
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_09_Simulate_Move_Birds: sdk=
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_10_Simulate_Move_Mammals: sd
[Robolectric] org.aplas.animaltour.TestB4MultimediaResources061.check_11_Simulate_Move_Inverts: sd

Process finished with exit code 0
```

You have to try until get all green checkes and continue to the next task.

You can get the screen display like below when running this Activity. The style of layout depends on your layout design.



My Animal Tour

Fishes

A group of cold-blooded vertebrates that breathe with gills rather than lungs, live in water, and generally lay eggs, although some bear their young alive.

Amphibians

A group of cold-blooded animals which can live on land or in water and who breathes with gills when young and with lungs as an adult.

Reptiles

A group of cold-blooded animals which have skins covered with small hard plates called scales and lay eggs.

Birds

A group of warm-blooded vertebrates distinguished by their feathers and their two legs and two wings.

Mammals

A group of warm-blooded animals that is distinguished by the possession of hair or fur, females that secrete milk for the nourishment of the young, and (typically) the birth of live young.

Invertebrates

Any animal that lacks a vertebral column, or backbone, in contrast to the cartilaginous or bony vertebrates. More than 90 percent of all living animal species are invertebrates.