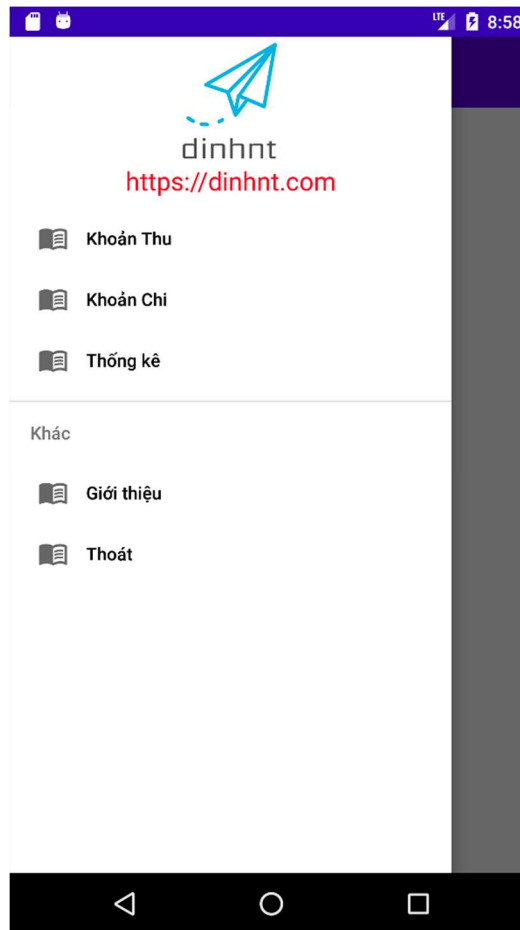


# FPL ASM AndroidUI (DinhNT)

Project Demo: [https://github.com/dinhntnguyenn/FPL\\_ASM\\_AndroidUI](https://github.com/dinhntnguyenn/FPL_ASM_AndroidUI)

## 1. Tạo DrawerLayout



- Tạo một số Fragment tương ứng với các chức năng trong Menu (Drawer Navigation)

Ví dụ ở đây tạo Fragment **Khoản thu**, các chức năng còn lại cũng tạo các Fragment tương tự

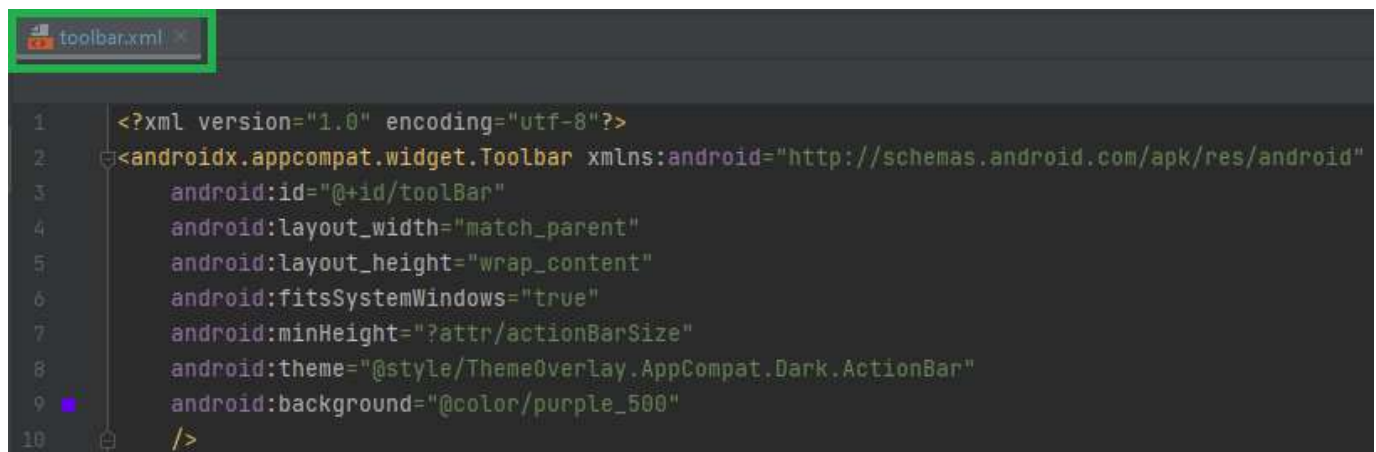
```
KhoanThuFragment.java
1 package com.dinhnt.fpl_asm_androidui.fragment.khoan_thu;
2
3 import ...
4
17 public class KhoanThuFragment extends Fragment {
18
19     @Nullable
20     @Override
21     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
22         View view = inflater.inflate(R.layout.fragment_khoan_thu, container, attachToRoot: false);
23
24         return view;
25     }
26 }
27 }
```

- Tạo 1 file menu và hình ảnh cho menu



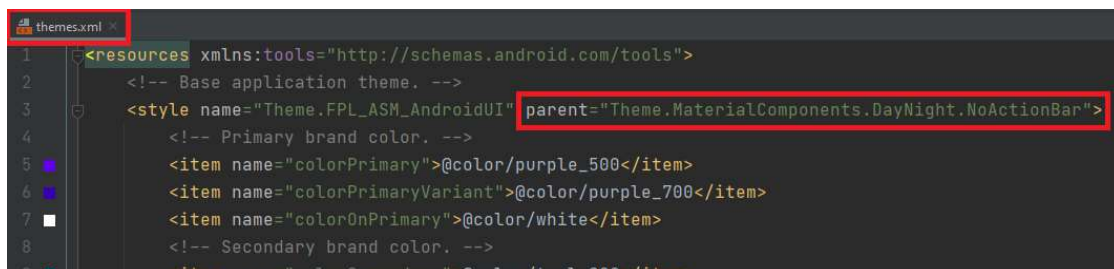
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3   <item
4     android:id="@+id/menuKhoanThu"
5     android:icon="@drawable/ic_action_name"
6     android:title="Khoản Thu" />
7
8   <item
9     android:id="@+id/menuKhoanChi"
10    android:icon="@drawable/ic_action_name"
11    android:title="Khoản Chi" />
12
13   <item
14     android:id="@+id/menuThongKe"
15     android:icon="@drawable/ic_action_name"
16     android:title="Thống kê" />
17
18   <item android:id="@+id/menuKhac"
19     android:title="Khác">
20     <menu>
21       <item
22         android:id="@+id/menuGioiThieu"
23         android:icon="@drawable/ic_action_name"
24         android:title="Giới thiệu" />
25
26       <item
27         android:id="@+id/menuThoat"
28         android:icon="@drawable/ic_action_name"
29         android:title="Thoát" />
30     </menu>
31   </item>
32
33 </menu>
```

- Tạo 1 file (layout) toolbar



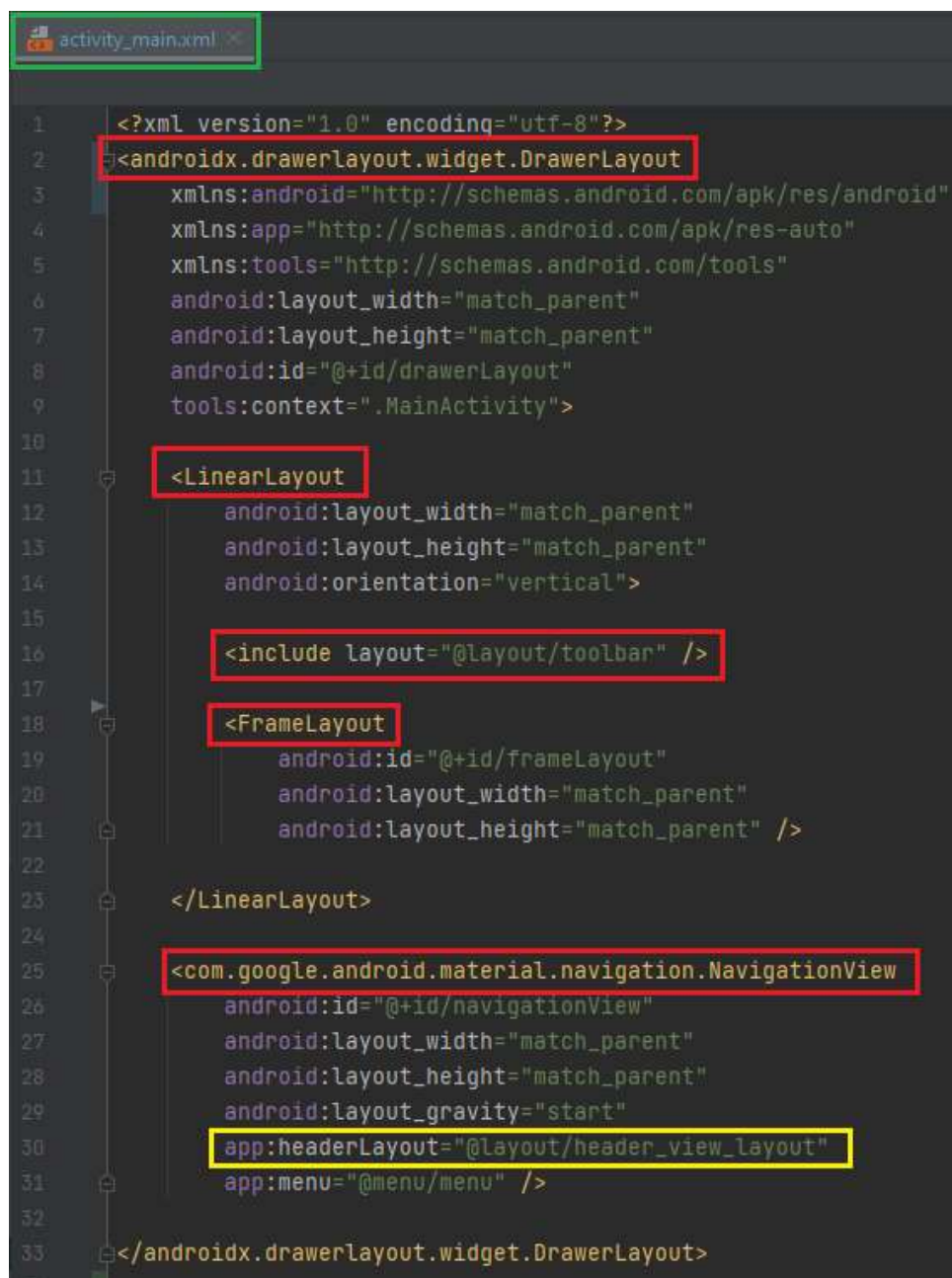
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.appcompat.widget.Toolbar xmlns:android="http://schemas.android.com/apk/res/android"
3   android:id="@+id/toolBar"
4   android:layout_width="match_parent"
5   android:layout_height="wrap_content"
6   android:fitsSystemWindows="true"
7   android:minHeight="?attr/actionBarSize"
8   android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
9   android:background="@color/purple_500"
10 />
```

- Sửa style thành NoActionBar trong file style để dùng Toolbar



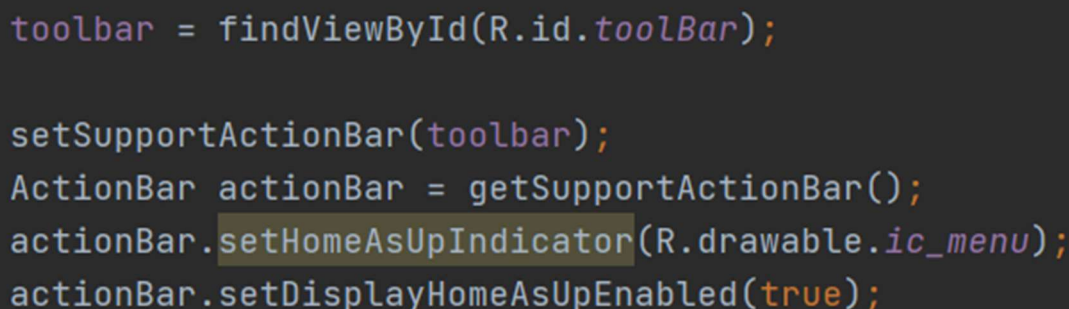
```
1 <resources xmlns:tools="http://schemas.android.com/tools">
2   <!-- Base application theme. -->
3   <style name="Theme.FPL_ASM_AndroidUI" parent="Theme.MaterialComponents.DayNight.NoActionBar">
4     <!-- Primary brand color. -->
5     <item name="colorPrimary">@color/purple_500</item>
6     <item name="colorPrimaryVariant">@color/purple_700</item>
7     <item name="colorOnPrimary">@color/white</item>
8     <!-- Secondary brand color. -->
9     <item name="colorSecondary">@color/teal_200</item>
```

- Trong layout chính (MainActivity) thiết kế giao diện gồm **DrawerLayout** làm container lớn nhất. Bên trong gồm **LinearLayout** cho phần giữa và **NavigationView**. Trong **LinearLayout**, gắn file layout **toolbar** đã làm ở trên và một **FrameLayout** để load các **Fragment** khi click bên **Navigation**.



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.drawerlayout.widget.DrawerLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      xmlns:app="http://schemas.android.com/apk/res-auto"
5      xmlns:tools="http://schemas.android.com/tools"
6      android:layout_width="match_parent"
7      android:layout_height="match_parent"
8      android:id="@+id/drawerLayout"
9      tools:context=".MainActivity">
10
11      <LinearLayout
12          android:layout_width="match_parent"
13          android:layout_height="match_parent"
14          android:orientation="vertical">
15
16          <include layout="@layout/toolbar" />
17
18          <FrameLayout
19              android:id="@+id/frameLayout"
20              android:layout_width="match_parent"
21              android:layout_height="match_parent" />
22
23      </LinearLayout>
24
25      <com.google.android.material.navigation.NavigationView
26          android:id="@+id/navigationView"
27          android:layout_width="match_parent"
28          android:layout_height="match_parent"
29          android:layout_gravity="start"
30          app:headerLayout="@layout/header_view_layout"
31          app:menu="@menu/menu" />
32
33  </androidx.drawerlayout.widget.DrawerLayout>
```

- Trong **MainActivity**, ánh xạ **Toolbar** và gắn menu lên **Toolbar**



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

setSupportActionBar(toolbar);
ActionBar actionBar = getSupportActionBar();
actionBar.setHomeAsUpIndicator(R.drawable.ic_menu);
actionBar.setDisplayHomeAsUpEnabled(true);
```

- Bắt sự kiện khi click icon trên ActionBar/ToolBar hiển thị trên Navigation Drawer bằng cách override hàm **onOptionsItemSelected** trong MainActivity.

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if (item.getItemId() == android.R.id.home) {
        drawerLayout.openDrawer(GravityCompat.START);
    }
    return super.onOptionsItemSelected(item);
}
```

- Tiếp theo đó, bắt sự kiện navigation.setNavigationItemSelectedListener để click vào các item trên Navigation Drawer sẽ chuyển các Fragment tương ứng.

```
navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        Fragment fragment;
        Class fragmentClass;
        switch (item.getItemId()) {
            case R.id.menuKhoanThu:
                fragmentClass = KhoanThuFragment.class;
                break;
            case R.id.menuKhoanChi:
                fragmentClass = KhoanChiFragment.class;
                break;
            case R.id.menuThongKe:
                fragmentClass = ThongKeFragment.class;
                break;
            default:
                fragmentClass = KhoanThuFragment.class;
                break;
        }

        try {
            fragment = (Fragment) fragmentClass.newInstance();
            FragmentManager fragmentManager = getSupportFragmentManager();
            fragmentManager.beginTransaction().replace(R.id.frameLayout, fragment).commit();

            //item.setChecked(true);
            setTitle(item.getTitle());
            drawerLayout.closeDrawer(GravityCompat.START);
        } catch (Exception exception) {
        }

        return false;
    }
});
```

## 2. Tạo file SQLiteOpenHelper (có thể tham khảo các bên dưới)

Tạo thêm 1 class tên QuanLyThuChiSQLite kế thừa từ **SQLiteOpenHelper**. Khai báo một số hằng số. Override 2 phương thức **onCreate**, **onUpgrade** và hàm tạo bằng cách chọn sửa lỗi nhanh. **Chú ý**, sau khi có hàm tạo nên sửa lại hàm tạo chỉ nhận 1 đối số là Context và trong hàm super của nó truyền tên database và version luôn để tí làm cho dễ.

```
public class QuanLyThuChiSQLite extends SQLiteOpenHelper {

    SQLiteDatabase db = this.getReadableDatabase();

    private static final String DATABASE_NAME = "quanlythuchi";
    private static final int DATABASE_VERSION = 1;

    private static final String TABLE_PHANLOAI = "phanloai";
    private static final String KEY_MALOAI = "maloi";
    private static final String KEY_TENLOAI = "tenloai";
    private static final String KEY_TRANGTHAI = "trangthai";

    private static final String TABLE_GIAODICH = "giaodich";
    private static final String KEY_MAGD = "magd";
    private static final String KEY_TIEUDE = "tieude";
    private static final String KEY_NGAY = "ngay";
    private static final String KEY_TIEN = "tien";
    private static final String KEY_MOTA = "mota";

    public QuanLyThuChiSQLite(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

    }

}
```



- Hàm onCreate

```
@Override
public void onCreate(SQLiteDatabase db) {
    String sqlPhanLoai = "create table " + TABLE_PHANLOAI +
        "(" +
        KEY_MALOA + " integer primary key autoincrement, " +
        KEY_TENLOAI + " text, " +
        KEY_TRANGTHAI + " text" +
        ")";
    db.execSQL(sqlPhanLoai);

    //tương tự cho table giao dich
}
```

- Hàm onUpgrade

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("drop table if exists " + TABLE_PHANLOAI);
    onCreate(db);

    //tương tự cho table giao dich
}
```

- Hàm Lấy danh sách loại khoản thu (làm tương tự cho loại khoản chi)

```
//lấy danh sách loại khoản thu
public ArrayList<LoaiThuChi> getAllLoaiKhoanThu() {
    Cursor cursor = db.rawQuery( sql: "select * from " + TABLE_PHANLOAI
        + " where " + KEY_TRANGTHAI + " = 'thu'", selectionArgs: null);

    ArrayList<LoaiThuChi> list = new ArrayList<>();
    if (cursor.getCount() != 0) {
        cursor.moveToFirst();
        do {
            list.add(new LoaiThuChi(cursor.getInt(0), cursor.getString(1)));
        } while (cursor.moveToNext());
    }
    return list;
}
```

- Hàm **Lấy thông tin loại khoản thu** (làm tương tự cho loại khoản chi)

```
//lấy thông tin loại khoản thu
public LoaiThuChi getThongTinLoaiKhoanThu(int maLoai) {
    Cursor cursor = db.rawQuery( sql: "select * from " + TABLE_PHANLOAI
    + " where " + KEY_MALOAI + " = " + maLoai, selectionArgs: null);
    return new LoaiThuChi(cursor.getInt( 0), cursor.getString( 1));
}
```

- Hàm **Tạo loại khoản thu mới** (làm tương tự cho loại khoản chi)

```
//Tạo loại khoản thu mới
public void taoLoaiKhoanThu(String tenLoai) {
    ContentValues values = new ContentValues();
    values.put(KEY_TENLOAI, tenLoai);
    values.put(KEY_TRANGTHAI, "thu");

    db.insert(TABLE_PHANLOAI, nullColumnHack: null, values);
}
```

- Hàm **Cập nhật loại khoản thu** (làm tương tự cho loại khoản chi)

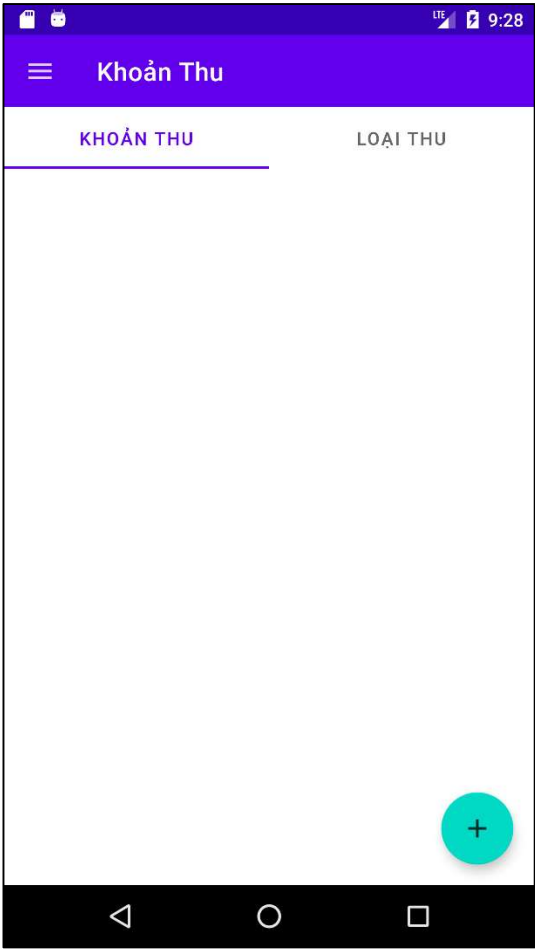
```
//Cập nhật loại khoản thu
public void capNhatLoaiKhoanThu(int maLoai, String tenLoai) {
    ContentValues values = new ContentValues();
    values.put(KEY_TENLOAI, tenLoai);
    values.put(KEY_TRANGTHAI, "thu");

    db.update(TABLE_PHANLOAI, values, whereClause: "maLoai=?", new String[]{String.valueOf(maLoai)});
}
```

- Hàm **Xóa loại khoản thu** (làm tương tự cho loại khoản chi)

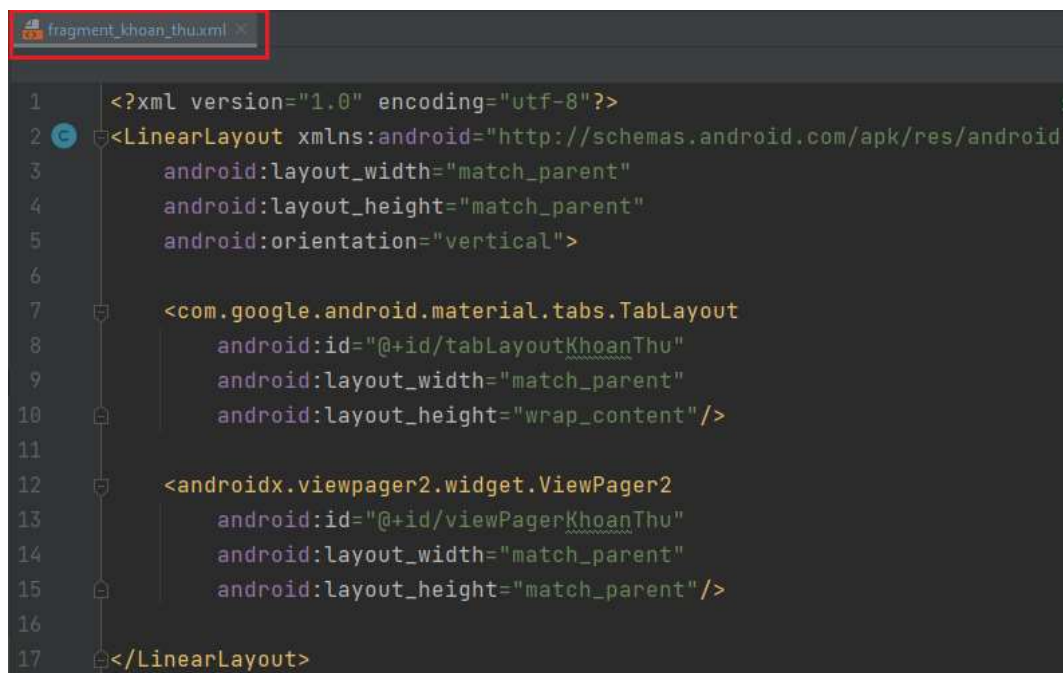
```
//Xóa loại khoản thu
public void xoaLoaiKhoanThu(int maLoai) {
    db.delete(TABLE_PHANLOAI, whereClause: "maLoai=?", new String[]{String.valueOf(maLoai)});
}
```

3. Tạo ViewPager kết hợp TabLayout cho Fragment **Khoản thu**





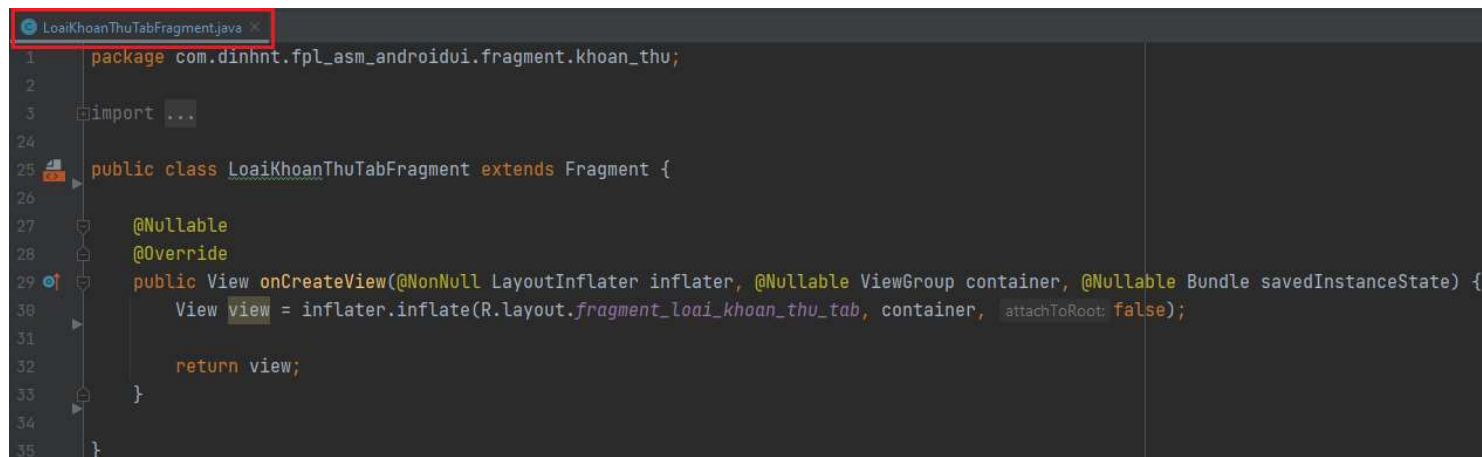
- Trong layout của **Fragment Khoản thu**



```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:orientation="vertical">
6
7     <com.google.android.material.tabs.TabLayout
8         android:id="@+id/tabLayoutKhoanThu"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"/>
11
12    <androidx.viewpager2.widget.ViewPager2
13        android:id="@+id/viewPagerKhoanThu"
14        android:layout_width="match_parent"
15        android:layout_height="match_parent"/>
16
17 </LinearLayout>
```

- Tạo các Fragment con tương ứng cho mỗi tab (**LoaiKhoanThuTabFragment**, **KhoanThuTabFragment**)

Ví dụ ở đây tạo fragment **LoaiKhoanThuTabFragment** là fragment cho tab **Loại thu** trong Fragment **Khoản thu**



```
1 package com.dinhnt.fpl_asm_androidui.fragment.khoan_thu;
2
3 import ...
4
24
25 public class LoaiKhoanThuTabFragment extends Fragment {
26
27     @Nullable
28     @Override
29     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
30         View view = inflater.inflate(R.layout.fragment_loai_khoan_thu_tab, container, attachToRoot: false);
31
32         return view;
33     }
34
35 }
```

- Tạo adapter cho Viewpager đặt tên là **ViewPagerKhoanThuAdapter**

Tạo constructor cho adapter với tham số truyền vào là **FragmentActivity**

**createFragment** truyền thứ tự vị trí của các fragment trên tablayout

**getItemCount** là số lượng tab trên tablayout

```
ViewPagerKhoanThuAdapter.java
1 package com.dinhnt.fpl_asm_androidui.adapter;
2
3 import ...
10
11 public class ViewPagerKhoanThuAdapter extends FragmentStateAdapter {
12
13     public ViewPagerKhoanThuAdapter(@NonNull FragmentActivity fragmentActivity) {
14         super(fragmentActivity);
15     }
16
17     @NonNull
18     @Override
19     public Fragment createFragment(int position) {
20         switch (position){
21             case 0:
22                 return new KhoanThuTabFragment();
23             case 1:
24                 return new LoaiKhoanThuTabFragment();
25             default:
26                 return new KhoanThuTabFragment();
27         }
28     }
29
30     @Override
31     public int getItemCount() { return 2; }
34 }
```

Trong **Fragment Khoản thu** ánh xạ ViewPager2, TabLayout và khởi tạo **ViewPagerKhoanThuAdapter** đã làm ở trên.

Gọi hàm **TabLayoutMediator** truyền vào 3 tham số là **tablayout**, **viewpager2** và **TabconfigurationStrategy** để set tiêu đề cho các tab trong TabLayout

```
KhoanThuFragment.java
17
18 public class KhoanThuFragment extends Fragment {
19
20     ViewPager2 viewPagerKhoanThu;
21     TabLayout tabLayoutKhoanThu;
22     ViewPagerKhoanThuAdapter adapter;
23
24     @Nullable
25     @Override
26     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
27         View view = inflater.inflate(R.layout.fragment_khoan_thu, container, attachToRoot: false);
28
29         viewPagerKhoanThu = view.findViewById(R.id.viewPagerKhoanThu);
30         tabLayoutKhoanThu = view.findViewById(R.id.tabLayoutKhoanThu);
31
32         adapter = new ViewPagerKhoanThuAdapter(getActivity());
33
34         viewPagerKhoanThu.setAdapter(adapter);
35
36         new TabLayoutMediator(tabLayoutKhoanThu, viewPagerKhoanThu, new TabLayoutMediator.TabConfigurationStrategy() {
37             @Override
38             public void onConfigureTab(@NonNull TabLayout.Tab tab, int position) {
39                 switch (position){
40                     case 0:
41                         tab.setText("Khoản thu");
42                         break;
43                     case 1:
44                         tab.setText("Loại thu");
45                         break;
46                 }
47             }
48         }).attach();
49
50         return view;
51     }
52 }
```

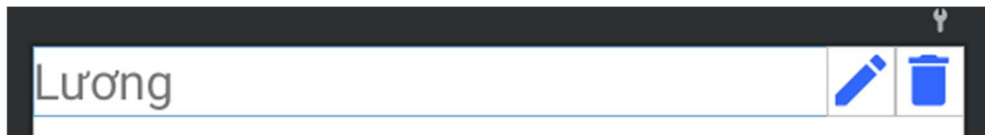
#### 4. Xây dựng layout và chức năng cho các Fragment con phục vụ TabLayout

- Thiết kế layout cho **LoaiKhoanThuTabFragment**

```
fragment_loai_khoan_thu_tab.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <androidx.coordinatorlayout.widget.CoordinatorLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      xmlns:app="http://schemas.android.com/apk/res-auto">
7
8      <ListView
9          android:id="@+id/listLoaiKhoanThu"
10         android:layout_width="match_parent"
11         android:layout_height="match_parent" />
12
13     <com.google.android.material.floatingactionbutton.FloatingActionButton
14         android:id="@+id/floatThemLoaiKhoanThu"
15         android:layout_width="wrap_content"
16         android:layout_height="wrap_content"
17         android:layout_gravity="bottom|right"
18         android:layout_margin="16dp"
19         android:src="@drawable/ic_add"
20         app:layout_anchor="@id/listLoaiKhoanThu"
21         app:layout_anchorGravity="bottom|right|end" />
22
23 </androidx.coordinatorlayout.widget.CoordinatorLayout>
```

- Xây dựng chức năng hiển thị danh sách các loại khoản thu (làm tương tự cho khoản chi)
  - o Xây dựng Adapter hiển thị danh sách trên ListView

Tạo item view cho từng item trên ListView



```
item_loai_khoan_thu.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:orientation="horizontal"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent">
6
7     <TextView
8         android:id="@+id/txtLoaiKhoanThu"
9         android:layout_width="match_parent"
10        android:layout_height="wrap_content"
11        android:textSize="22sp"
12        android:layout_toStartOf="@id/imgEdit"/>
13
14
15     <ImageView
16         android:id="@+id/imgEdit"
17         android:layout_width="30dp"
18         android:layout_height="30dp"
19         android:layout_toStartOf="@id/imgDelete"
20         android:src="@drawable/ic_edit" />
21
22     <ImageView
23         android:id="@+id/imgDelete"
24         android:layout_width="30dp"
25         android:layout_height="30dp"
26         android:src="@drawable/ic_delete"
27         android:layout_alignParentEnd="true"
28         />
29
30 </RelativeLayout>
```



Tạo adapter **LoaiKhoanThuAdapter** extends **BaseAdapter**

Tạo **constructor** cho adapter truyền vào 3 tham số: danh sách loại thu chi, context và SQLiteOpenHelper (QuanLyThuChiSQLite)

**getCount**: tổng số loại khoản thu

**getItem**: lấy thông tin loại khoản thu theo vị trí

**getItemId**: lấy id loại khoản thu theo vị trí

```
public class LoaiKhoanThuAdapter extends BaseAdapter {

    ArrayList<LoaiThuChi> alThuChi = new ArrayList<>();
    Context context;
    QuanLyThuChiSQLite quanLyThuChi;

    public LoaiKhoanThuAdapter(ArrayList<LoaiThuChi> alThuChi, Context context, QuanLyThuChiSQLite quanLyThuChi) {
        this.alThuChi = alThuChi;
        this.context = context;
        this.quanLyThuChi = quanLyThuChi;
    }

    @Override
    public int getCount() {
        return alThuChi.size();
    }

    @Override
    public Object getItem(int i) {
        return alThuChi.get(i);
    }

    @Override
    public long getItemId(int i) {
        return alThuChi.get(i).getMaLoai();
    }
}
```

Tạo class **ViewOfItem** để tối ưu hóa việc render các item trên ListView

**getView**: ánh xạ, hiển thị thông tin và thực hiện các chức năng cho mỗi item trong ListView

```
class ViewOfItem{
    TextView txtLoaiKhoanThu;
    ImageView imgEdit, imgDelete;
}

@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    ViewOfItem viewOfItem;
    LayoutInflater inflater = ((Activity)context).getLayoutInflater();

    if(view == null){
        view = inflater.inflate(R.layout.item_loai_khoan_thu, root: null);
        viewOfItem = new ViewOfItem();
        viewOfItem.txtLoaiKhoanThu = view.findViewById(R.id.txtLoaiKhoanThu);
        viewOfItem.imgEdit = view.findViewById(R.id.imgEdit);
        viewOfItem.imgDelete = view.findViewById(R.id.imgDelete);

        view.setTag(viewOfItem);
    }else{
        viewOfItem = (ViewOfItem) view.getTag();
    }

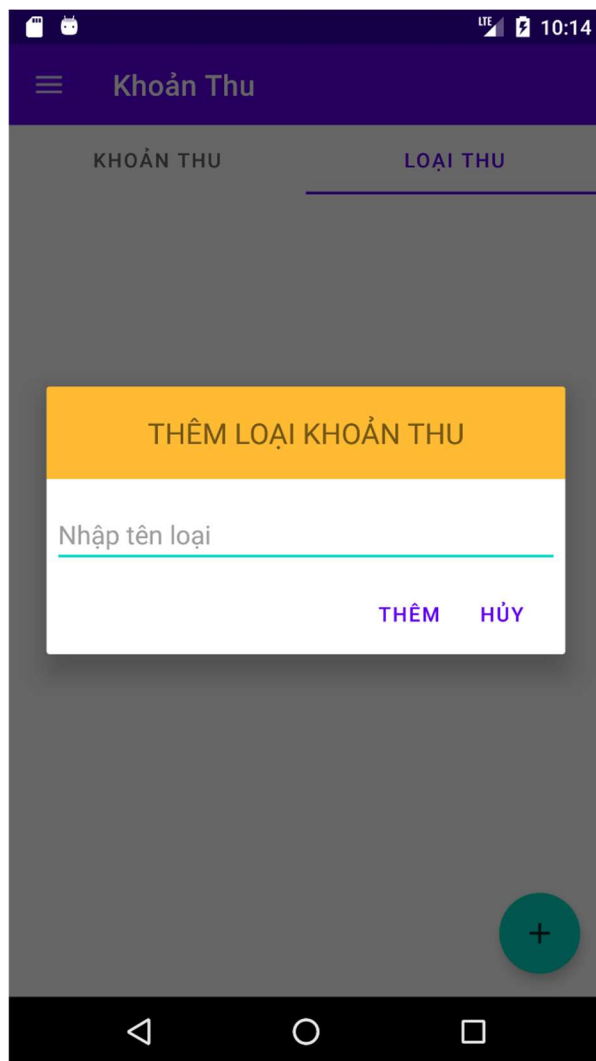
    viewOfItem.txtLoaiKhoanThu.setText(alThuChi.get(i).getTenLoai());

    return view;
}
```

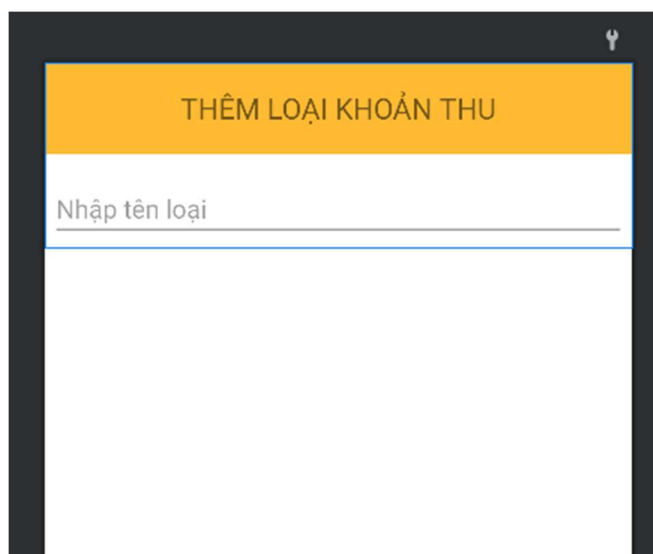
## Hiển thị danh sách các loại khoản thu

```
LoaiKhoanThuTabFragment.java
24
25 public class LoaiKhoanThuTabFragment extends Fragment {
26
27     ListView listLoaiKhoanThu;
28     FloatingActionButton floatThemLoaiKhoanThu;
29     QuanLyThuChiSQLite quanLyThuChi;
30     LoaiKhoanThuAdapter adapter;
31     ArrayList<LoaiThuChi> alLoaiThu;
32
33     @Nullable
34     @Override
35     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
36         View view = inflater.inflate(R.layout.fragment_loai_khoan_thu_tab, container, attachToRoot: false);
37
38         listLoaiKhoanThu = view.findViewById(R.id.listLoaiKhoanThu);
39         floatThemLoaiKhoanThu = view.findViewById(R.id.floatThemLoaiKhoanThu);
40
41         quanLyThuChi = new QuanLyThuChiSQLite(getActivity());
42         alLoaiThu = new ArrayList<>();
43
44         getDataLoaiThu();
45
46         return view;
47     }
48
49     private void getDataLoaiThu() {
50         alLoaiThu.clear();
51         alLoaiThu = quanLyThuChi.getAllLoaiKhoanThu();
52         adapter = new LoaiKhoanThuAdapter(alLoaiThu, getActivity(), quanLyThuChi);
53         listLoaiKhoanThu.setAdapter(adapter);
54         adapter.notifyDataSetChanged();
55     }
56 }
```

- Xây dựng **chức năng thêm** loại khoản thu mới (làm tương tự cho khoản chi)



- Thiết kế layout cho AlertDialog **thêm loại khoản thu mới**



```
dialog_them_loai_khoan_thu.xml ×  
  
1 <?xml version="1.0" encoding="utf-8"?>  
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
3     android:layout_width="match_parent"  
4     android:layout_height="wrap_content"  
5     android:orientation="vertical" >  
6  
7     <TextView  
8         android:id="@+id/title"  
9         android:layout_width="match_parent"  
10        android:layout_height="64dp"  
11        android:background="#FFFFBB33"  
12        android:gravity="center"  
13        android:text="THÊM LOẠI KHOẢN THU"  
14        android:textSize="20sp" />  
15  
16     <EditText  
17         android:id="@+id/edtTenLoaiKhoanThu"  
18         android:layout_width="match_parent"  
19         android:layout_height="wrap_content"  
20         android:layout_marginBottom="4dp"  
21         android:layout_marginLeft="4dp"  
22         android:layout_marginRight="4dp"  
23         android:layout_marginTop="16dp"  
24         android:hint="Nhập tên loại"  
25         android:inputType="textEmailAddress" />  
26 </LinearLayout>
```



Trong **LoaiKhoanThuTabFragment**

tạo thêm hàm **createDialogThemLoaiKhoanThu** để hiển thị AlertDialog thêm loại khoản thu

```
private void createDialogThemLoaiKhoanThu() {
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    LayoutInflater inflater = getLayoutInflater();
    View v = inflater.inflate(R.layout.dialog_them_loai_khoan_thu, root: null);

    EditText edtTenLoaiKhoanThu = v.findViewById(R.id.edtTenLoaiKhoanThu);

    builder.setView(v);

    builder.setNegativeButton(text: "Thêm", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            quanLyThuChi.taoLoaiKhoanThu(edtTenLoaiKhoanThu.getText().toString());
            getDataLoaiThu();
        }
    });

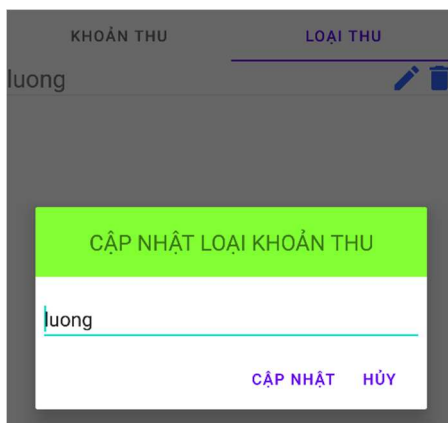
    builder.setPositiveButton(text: "Hủy", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.cancel();
        }
    });

    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
```

Gọi sự kiện click vào **FloatingActionButton** để mở hộp thoại thêm mới một loại khoản thu

```
floatThemLoaiKhoanThu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        createDialogThemLoaiKhoanThu();
    }
});
```

- Xây dựng **chức năng chỉnh sửa** loại khoản thu (làm tương tự cho khoản chi)



Thiết kế layout cho AlertDialog **cập nhật loại khoản thu** (tương tự như layout **thêm loại khoản thu mới**)

Trong adapter **LoaiKhoanThuAdapter** viết thêm hàm **createDialogCapNhatLoaiKhoanThu** để hiển thị hộp thoại cập nhật thông tin loại khoản thu

```
private void createDialogCapNhatLoaiKhoanThu(int maLoai, String tenLoai) {
    AlertDialog.Builder builder = new AlertDialog.Builder(context);
    LayoutInflater inflater = ((Activity)context).getLayoutInflater();
    View v = inflater.inflate(R.layout.dialog_cap_nhat_loai_khoan_thu, root: null);

    EditText edtTenLoaiKhoanThu = v.findViewById(R.id.edtTenLoaiKhoanThu);

    edtTenLoaiKhoanThu.setText(tenLoai);

    builder.setView(v);

    builder.setNegativeButton(text: "Cập nhật", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            quanLyThuChi.capNhatLoaiKhoanThu(maLoai, edtTenLoaiKhoanThu.getText().toString());
            loadDanhSach();
        }
    });

    builder.setPositiveButton(text: "Hủy", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.cancel();
        }
    });

    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
```

Viết thêm hàm **loadDanhSach** để load lại danh sách loại khoản thu sau khi cập nhật/xóa

```
private void loadDanhSach(){
    alThuChi.clear();
    alThuChi = quanLyThuChi.getAllLoaiKhoanThu();
    notifyDataSetChanged();
}
```

Viết sự kiện click vào image Edit hiển thị hộp thoại cập nhật thông tin

```
//cập nhật thông tin
viewOfItem.imgEdit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        createDialogCapNhatLoaiKhoanThu(alThuChi.get(i).getMaLoai(), alThuChi.get(i).getTenLoai());
    }
});
```

- Xây dựng **chức năng xóa** loại khoản thu (làm tương tự cho khoản chi)

Viết sự kiện click vào image Delete xóa loại khoản thu

```
//xóa
viewOfItem.imgDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        quanLyThuChi.xoaLoaiKhoanThu(alThuChi.get(i).getMaLoai());
        loadDanhSach();
    }
});
```

**Bonus:** Trong chức năng **thêm/chỉnh sửa khoản thu** có sử dụng spinner để load ra loại khoản thu, chúng ta làm như sau (làm tương tự cho khoản chi):

Viết hàm lấy ra danh sách các loại khoản thu và hiển thị lên spinner

```
private void getDataLoaiThu(Spinner sp) {
    alLoaiThu.clear();
    alLoaiThu = quanLyThuChi.getAllLoaiKhoanThu();

    ArrayList<HashMap<String, String>> list = new ArrayList<>();
    for (LoaiThuChi loaiThuChi : alLoaiThu) {
        HashMap<String, String> hashMap = new HashMap<>();
        hashMap.put("maLoai", String.valueOf(loaiThuChi.getMaLoai()));
        hashMap.put("tenLoai", loaiThuChi.getTenLoai());
        list.add(hashMap);
    }

    SimpleAdapter adapter = new SimpleAdapter(getActivity(),
        list, R.layout.item_spinner_loai_khoan_thu,
        new String[]{"tenLoai"}, new int[]{R.id.txtLoaiKhoanThu});
    sp.setAdapter(adapter);
}
```

Viết thêm hàm **createDialogThemKhoanThu** để hiển thị hộp thoại thêm khoản thu mới

```
private void createDialogThemKhoanThu() {
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    LayoutInflater inflater = getLayoutInflater();
    View v = inflater.inflate(R.layout.dialog_them_khoan_thu, root: null);

    //....
    Spinner spnLoaiKhoanThu = v.findViewById(R.id.spnLoaiKhoanThu);

    getDataLoaiThu(spnLoaiKhoanThu);

    builder.setView(v);

    builder.setNegativeButton( text: "Thêm", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            //...code

            HashMap<String, String> hashMap = (HashMap<String, String>) spnLoaiKhoanThu.getSelectedItem();
            Toast.makeText(getActivity(), text: "Mã loại: " + hashMap.get("maLoai"), Toast.LENGTH_SHORT).show();
        }
    });

    builder.setPositiveButton( text: "Hủy", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialogInterface, int i) {
            dialogInterface.cancel();
        }
    });

    AlertDialog alertDialog = builder.create();
    alertDialog.show();
}
```

lấy mã loại khoản thu tương ứng khi người dùng chọn trên spinner

Viết sự kiện click vào **FloatingActionButton** để hiển thị hộp thoại thêm khoản thu mới

```
floatThemKhoanThu.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        createDialogThemKhoanThu();
    }
});
```

CHÚC CÁC BẠN HOÀN THÀNH ASM THẬT TỐT <3

DinhNT