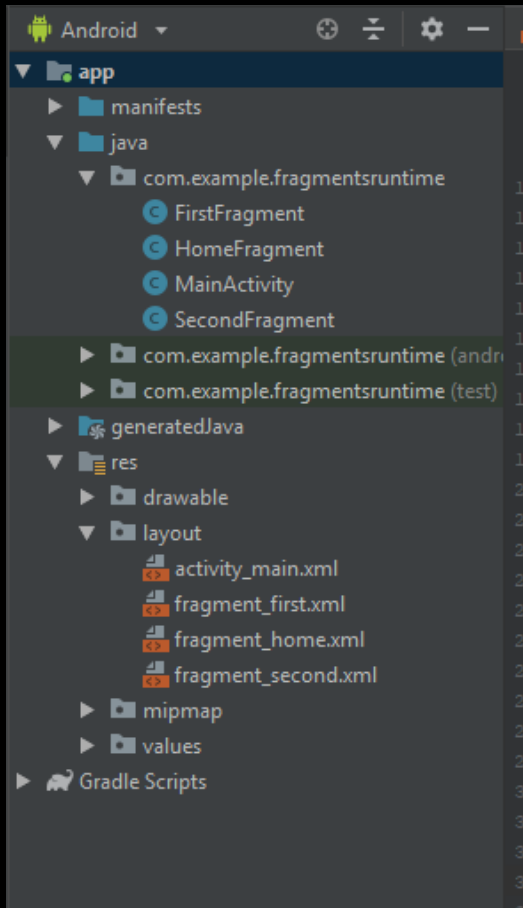


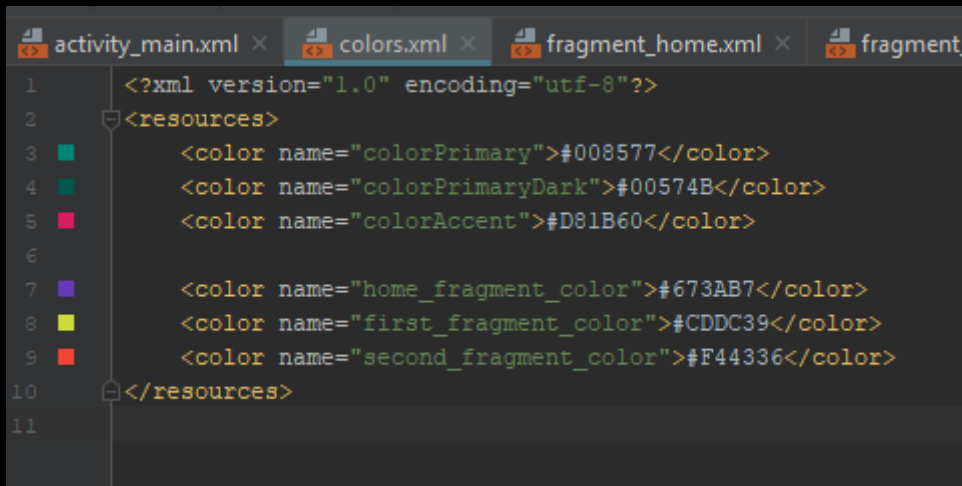
Przełączanie między fragmentami

Krok 1

Stwórz nowy projekt oraz 3 fragmenty.



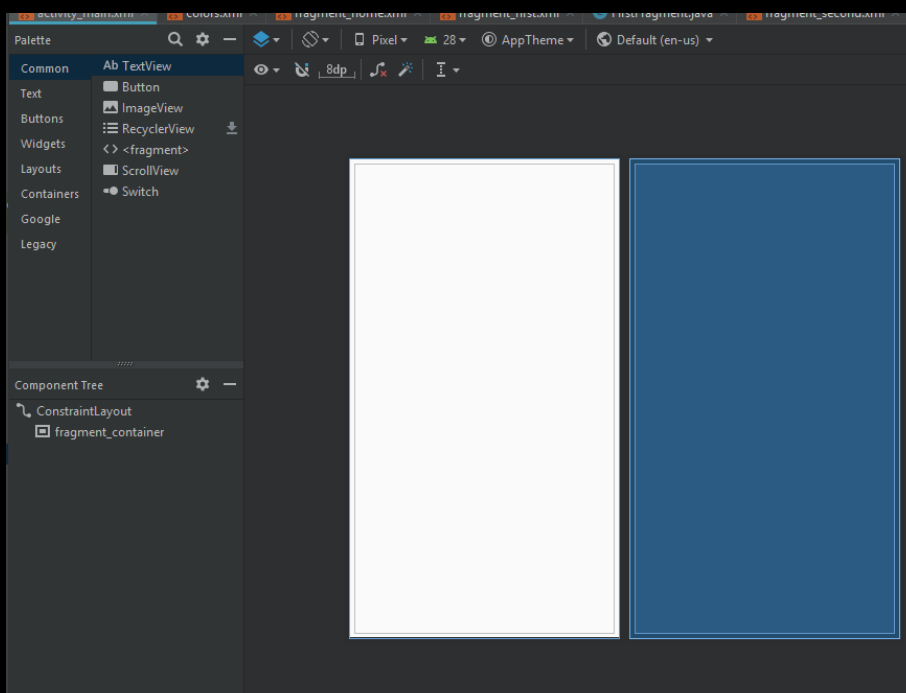
W pliku colors.xml dodaj 3 nowe kolory:



Krok 2

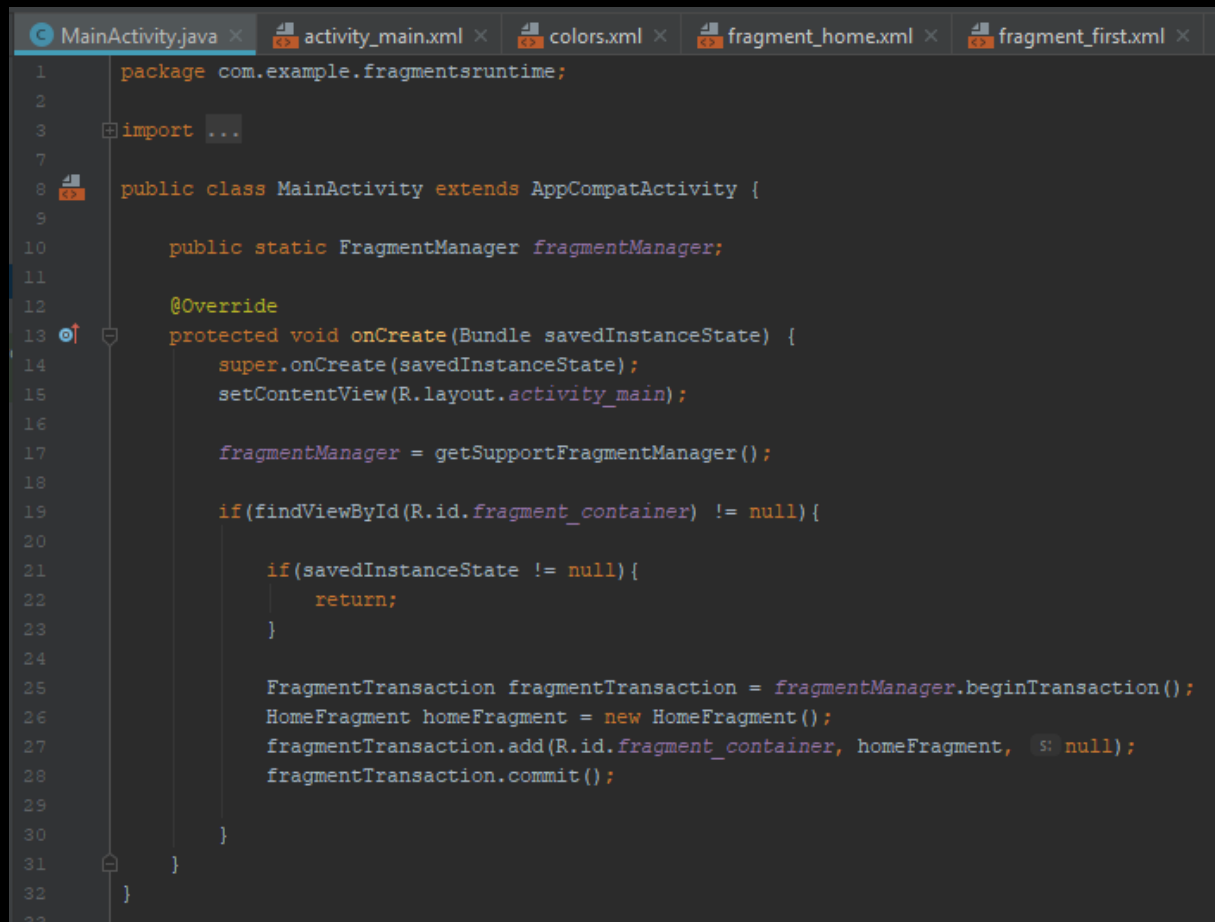
Zaprojektuj interfejs MainActivity. Stworzymy container do trzymania fragmentów wykorzystując `FrameLayout` (zwróć uwagę na id elementów!):

```
activity_main.xml × colors.xml × fragment_home.xml × fragment_first.x
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res-auto"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".MainActivity">
8
9      <FrameLayout
10         android:id="@+id/fragment_container"
11         android:layout_width="395dp"
12         android:layout_height="715dp"
13         android:layout_marginStart="8dp"
14         android:layout_marginTop="8dp"
15         android:layout_marginEnd="8dp"
16         android:layout_marginBottom="8dp"
17         app:layout_constraintBottom_toBottomOf="parent"
18         app:layout_constraintEnd_toEndOf="parent"
19         app:layout_constraintStart_toStartOf="parent"
20         app:layout_constraintTop_toTopOf="parent">
21
22     </FrameLayout>
23
24 </android.support.constraint.ConstraintLayout>
25
```



Krok 3

Musimy zaimplementować w MainActivity mechanizm wrzucenia do kontenera naszego pierwszego fragmentu:

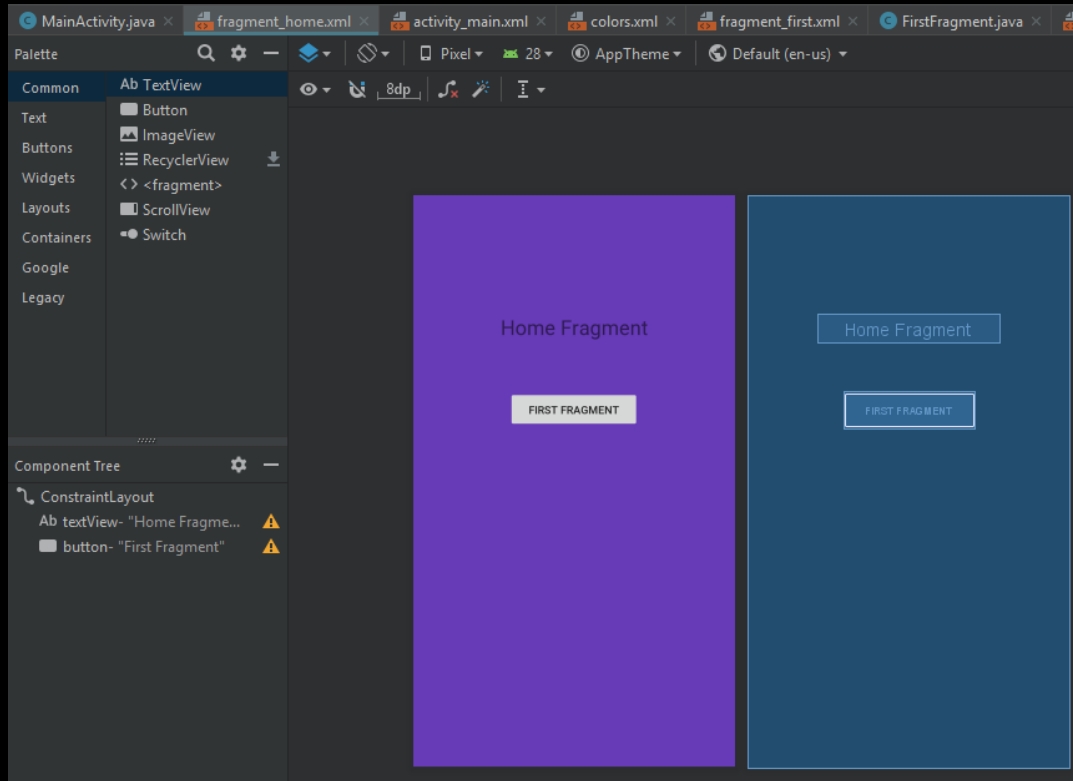


```
1 package com.example.fragmentsruntime;
2
3 import ...
4
5
6
7
8 public class MainActivity extends AppCompatActivity {
9
10     public static FragmentManager fragmentManager;
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         fragmentManager = getSupportFragmentManager();
18
19         if(findViewById(R.id.fragment_container) != null){
20
21             if(savedInstanceState != null){
22                 return;
23             }
24
25             FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();
26             HomeFragment homeFragment = new HomeFragment();
27             fragmentTransaction.add(R.id.fragment_container, homeFragment, null);
28             fragmentTransaction.commit();
29
30         }
31     }
32 }
33
```

Krok 4

Teraz projektujemy wygląd naszych fragmentów. Będzie to prosty TextView oraz przycisk.

HomeFragment:



MainActivity.java

fragment_home.xml

activity_main.xml

colors.xml

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

<android.support.constraint.ConstraintLayout

xmlns:android="http://schemas.android.com/apk/res/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=".HomeFragment"

android:background="@color/home_fragment_color">

<TextView

android:id="@+id/textView"

android:layout_width="233dp"

android:layout_height="36dp"

android:layout_marginStart="8dp"

android:layout_marginTop="8dp"

android:layout_marginEnd="8dp"

android:layout_marginBottom="8dp"

android:text="Home Fragment"

android:textAlignment="center"

android:textSize="26sp"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent"

app:layout_constraintVertical_bias="0.211" />

<Button

android:id="@+id/button"

android:layout_width="167dp"

android:layout_height="wrap_content"

android:layout_marginStart="8dp"

android:layout_marginTop="8dp"

android:layout_marginEnd="8dp"

android:layout_marginBottom="8dp"

android:text="First Fragment"

app:layout_constraintBottom_toBottomOf="parent"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintHorizontal_bias="0.498"

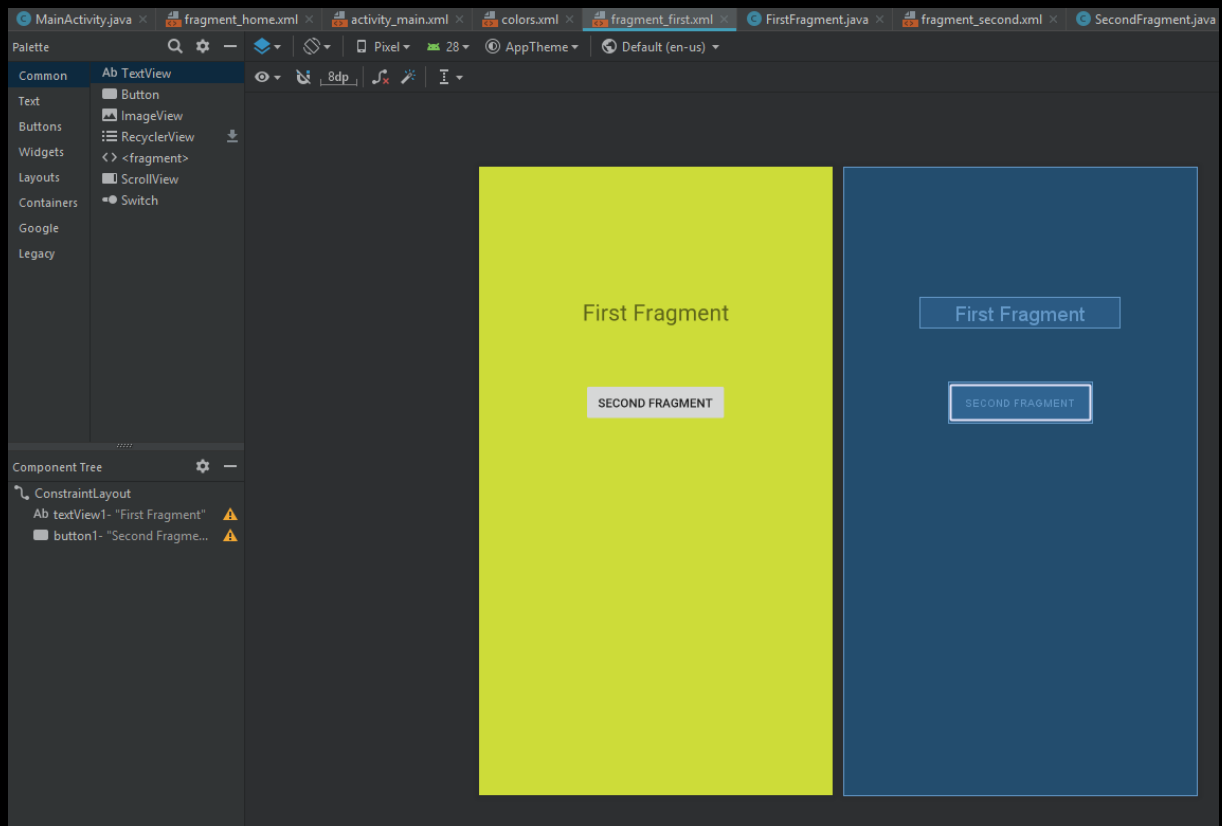
app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@+id/textView"

app:layout_constraintVertical_bias="0.114" />

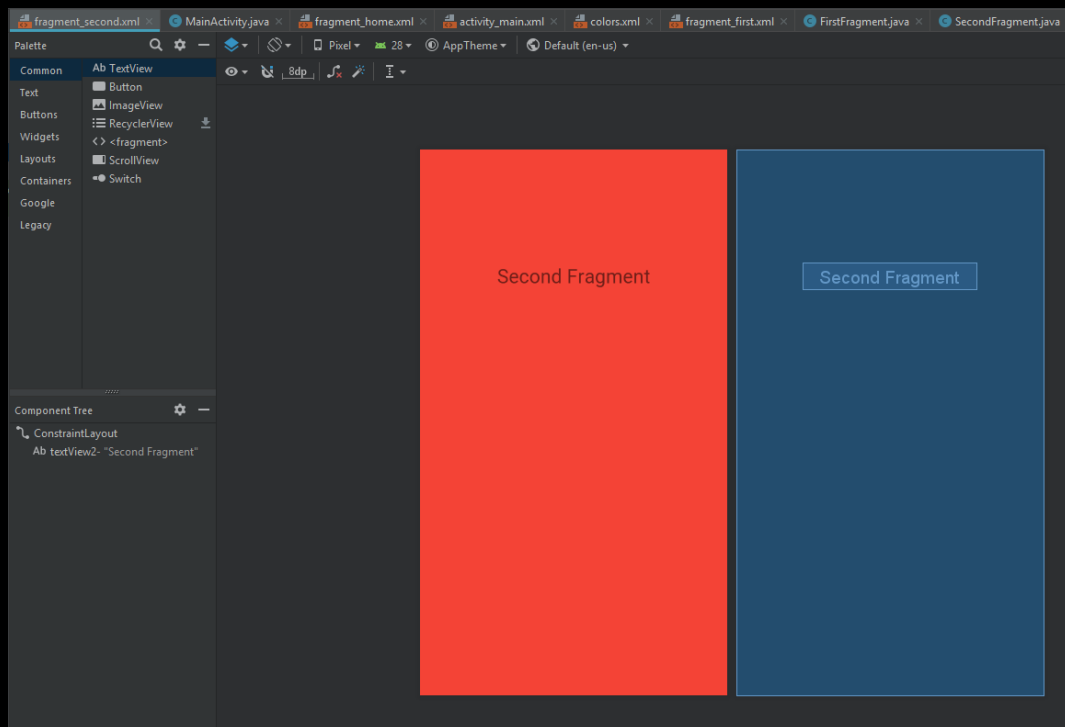
</android.support.constraint.ConstraintLayout>

FirstFragment:



```
MainActivity.java × fragment_home.xml × activity_main.xml × colors.xml ×
1      <?xml version="1.0" encoding="utf-8"?>
2      <android.support.constraint.ConstraintLayout
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          tools:context=".SecondFragment"
9          android:background="@color/first_fragment_color">
10
11      <TextView
12          android:id="@+id/textView1"
13          android:layout_width="233dp"
14          android:layout_height="36dp"
15          android:layout_marginStart="8dp"
16          android:layout_marginTop="8dp"
17          android:layout_marginEnd="8dp"
18          android:layout_marginBottom="8dp"
19          android:text="First Fragment"
20          android:textAlignment="center"
21          android:textSize="26sp"
22          app:layout_constraintBottom_toBottomOf="parent"
23          app:layout_constraintEnd_toEndOf="parent"
24          app:layout_constraintStart_toStartOf="parent"
25          app:layout_constraintTop_toTopOf="parent"
26          app:layout_constraintVertical_bias="0.211" />
27
28      <Button
29          android:id="@+id/button1"
30          android:layout_width="167dp"
31          android:layout_height="wrap_content"
32          android:layout_marginStart="8dp"
33          android:layout_marginTop="8dp"
34          android:layout_marginEnd="8dp"
35          android:layout_marginBottom="8dp"
36          android:text="Second Fragment"
37          app:layout_constraintBottom_toBottomOf="parent"
38          app:layout_constraintEnd_toEndOf="parent"
39          app:layout_constraintHorizontal_bias="0.498"
40          app:layout_constraintStart_toStartOf="parent"
41          app:layout_constraintTop_toBottomOf="@+id/textView1"
42          app:layout_constraintVertical_bias="0.114" />
43      </android.support.constraint.ConstraintLayout>
44
```

SecondFragment:

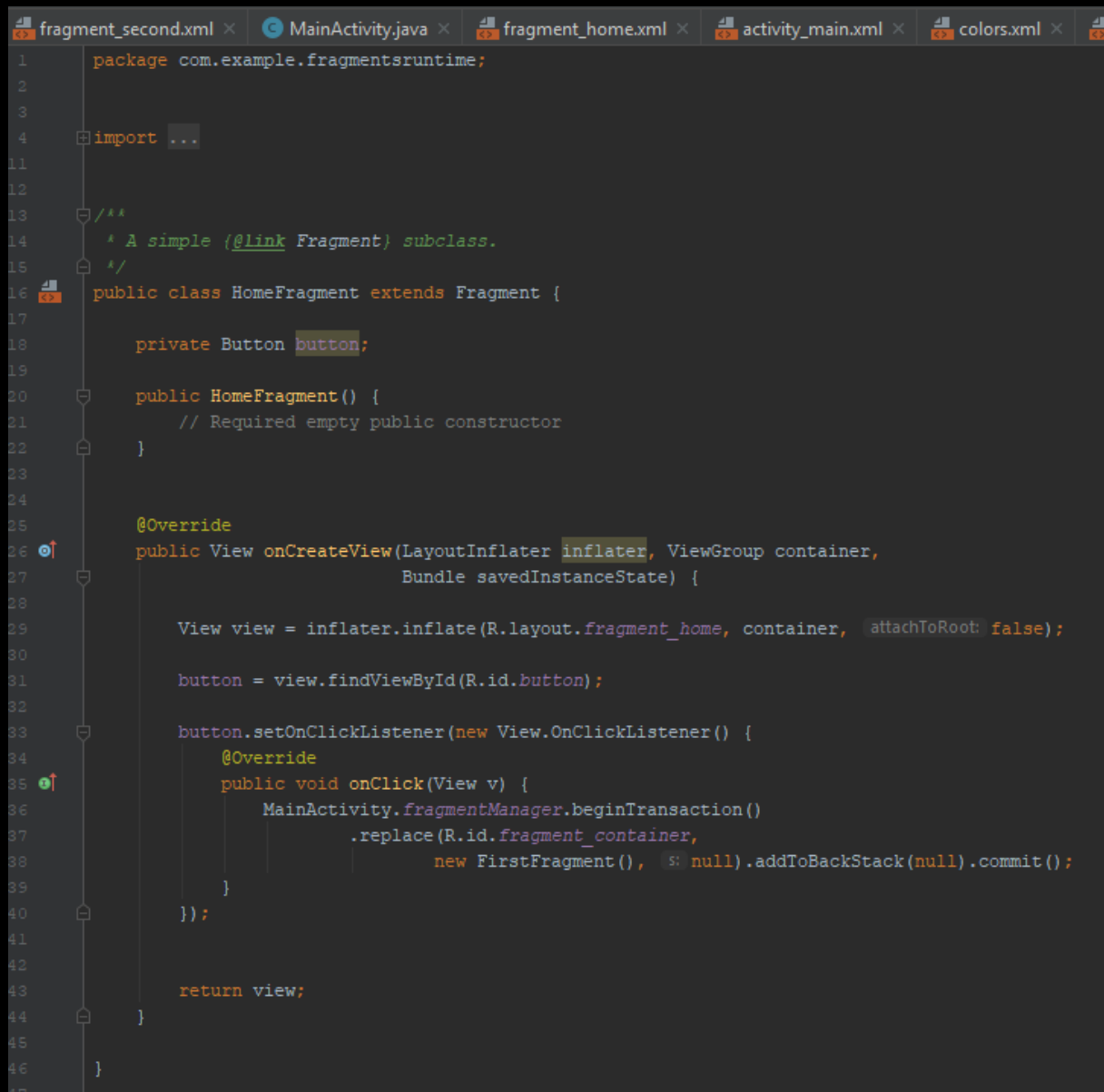


```
1  <?xml version="1.0" encoding="utf-8"?>
2  <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.and
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      tools:context=".SecondFragment"
8      android:background="@color/second_fragment_color">
9
10     <TextView
11         android:id="@+id/textView2"
12         android:layout_width="233dp"
13         android:layout_height="36dp"
14         android:layout_marginStart="8dp"
15         android:layout_marginTop="8dp"
16         android:layout_marginEnd="8dp"
17         android:layout_marginBottom="8dp"
18         android:text="Second Fragment"
19         android:textAlignment="center"
20         android:textSize="26sp"
21         app:layout_constraintBottom_toBottomOf="parent"
22         app:layout_constraintEnd_toEndOf="parent"
23         app:layout_constraintStart_toStartOf="parent"
24         app:layout_constraintTop_toTopOf="parent"
25         app:layout_constraintVertical_bias="0.211" />
26
27
28     </android.support.constraint.ConstraintLayout>
```


Krok 5

Teraz musimy dodać trochę kodu do naszych fragmentów, szczególnie znane Ci powinno być nasłuchiwanie naciśnięcia przycisku. Inaczej niż wywołanie nowej aktywności wygląda wywołanie nowego fragmentu. Bardziej można to nazwać zastępowaniem jednego fragmentu drugim w tym samym kontenerze oraz dodanie go do stosu by można było wrócić do poprzedniego fragmentu.

HomeFragment:



```
1 package com.example.fragmentsruntime;
2
3
4 import ...
5
6
7
8
9
10
11
12
13 /**
14  * A simple {@link Fragment} subclass.
15  */
16 public class HomeFragment extends Fragment {
17
18     private Button button;
19
20     public HomeFragment() {
21         // Required empty public constructor
22     }
23
24
25     @Override
26     public View onCreateView(LayoutInflater inflater, ViewGroup container,
27                             Bundle savedInstanceState) {
28
29         View view = inflater.inflate(R.layout.fragment_home, container, attachToRoot: false);
30
31         button = view.findViewById(R.id.button);
32
33         button.setOnClickListener(new View.OnClickListener() {
34             @Override
35             public void onClick(View v) {
36                 MainActivity.fragmentManager.beginTransaction()
37                     .replace(R.id.fragment_container,
38                             new FirstFragment(), null).addToBackStack(null).commit();
39             }
40         });
41
42         return view;
43     }
44 }
45
46 }
```

FirstFragment:

```
fragment_second.xml x MainActivity.java x fragment_home.xml x activity_main.xml x colors.xml x frag
1 package com.example.fragmentsruntime;
2
3
4 import ...
10
11
12 /**
13  * A simple {@link Fragment} subclass.
14  */
15 public class FirstFragment extends Fragment {
16
17     private Button button;
18
19     public FirstFragment() {
20         // Required empty public constructor
21     }
22
23
24     @Override
25     public View onCreateView(LayoutInflater inflater, ViewGroup container,
26                             Bundle savedInstanceState) {
27         // Inflate the layout for this fragment
28         View view = inflater.inflate(R.layout.fragment_first, container, attachToRoot: false);
29
30         button = view.findViewById(R.id.button1);
31
32         button.setOnClickListener(new View.OnClickListener() {
33             @Override
34             public void onClick(View v) {
35                 MainActivity.fragmentManager.beginTransaction()
36                     .replace(R.id.fragment_container,
37                             new SecondFragment(), null).addToBackStack(null).commit();
38             }
39         });
40
41         return view;
42     }
43
44 }
45
46
47
```

Uruchom aplikację.

