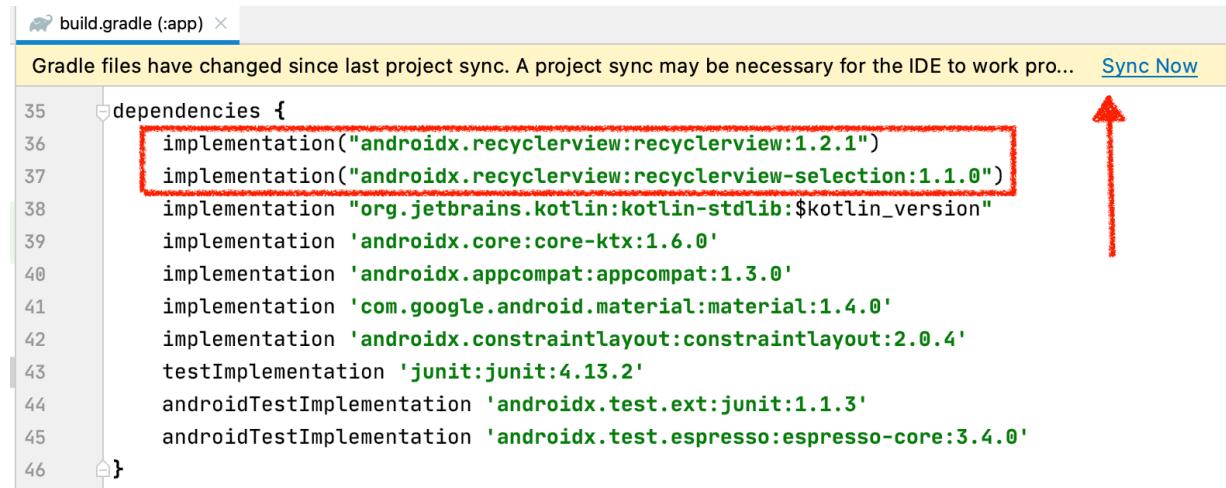


Obsługa RecyclerView – Przykład 2 z siatką typu Grid

Tworzymy nowy projekt o dowolnej nazwie z pustą aktywnością.

Dodajemy biblioteki RecyclerView do pliku build.gradle (Modules):

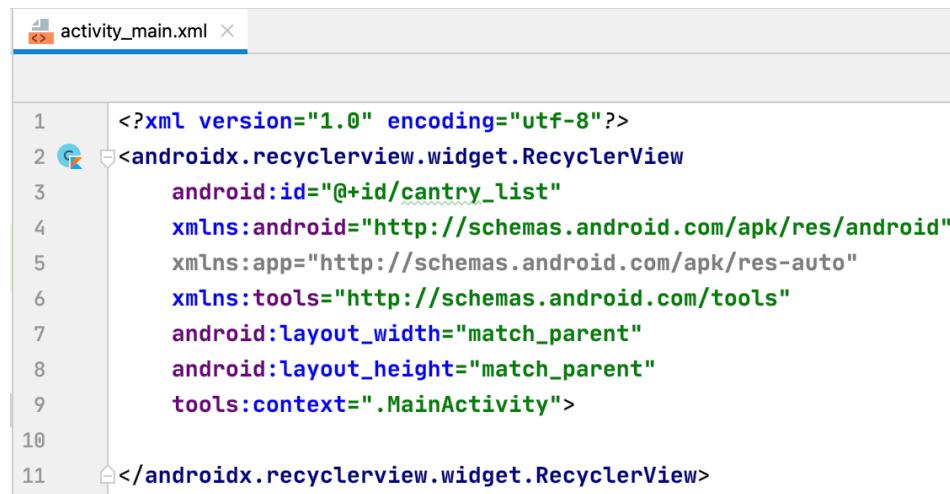


```

35 dependencies {
36     implementation("androidx.recyclerview:recyclerview:1.2.1")
37     implementation("androidx.recyclerview:recyclerview-selection:1.1.0") highlighted
38     implementation "org.jetbrains.kotlin:kotlin-stdlib:$kotlin_version"
39     implementation 'androidx.core:core-ktx:1.6.0'
40     implementation 'androidx.appcompat:appcompat:1.3.0'
41     implementation 'com.google.android.material:material:1.4.0'
42     implementation 'androidx.constraintlayout:constraintlayout:2.0.4'
43     testImplementation 'junit:junit:4.13.2'
44     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
45     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
46 }

```

W głównej aktywności zmieniamy nasz layout na RecyclerView

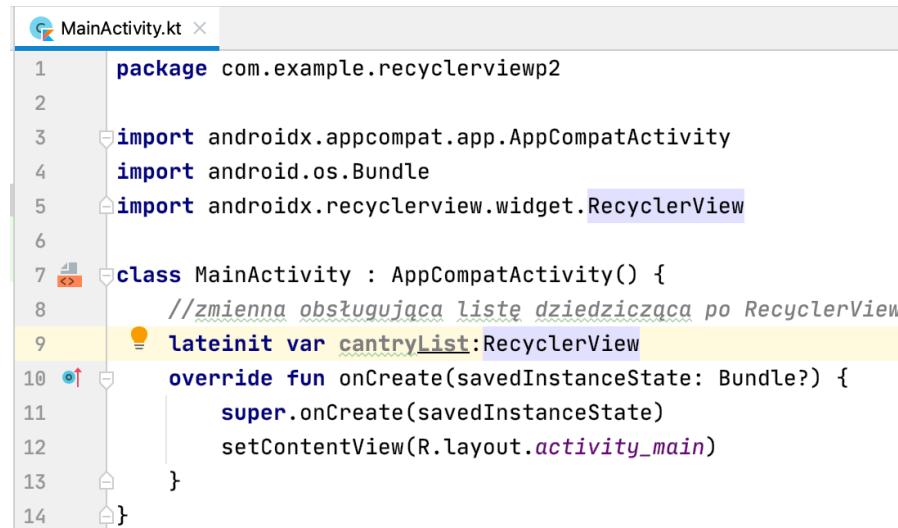


```

1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.recyclerview.widget.RecyclerView
3     android:id="@+id/cantry_list"
4     xmlns:android="http://schemas.android.com/apk/res/android"
5     xmlns:app="http://schemas.android.com/apk/res-auto"
6     xmlns:tools="http://schemas.android.com/tools"
7     android:layout_width="match_parent"
8     android:layout_height="match_parent"
9     tools:context=".MainActivity">
10
11 </androidx.recyclerview.widget.RecyclerView>

```

W pliku MainActivity.kt dodajemy zmienną dziedziczącą po klasie RecyclerView

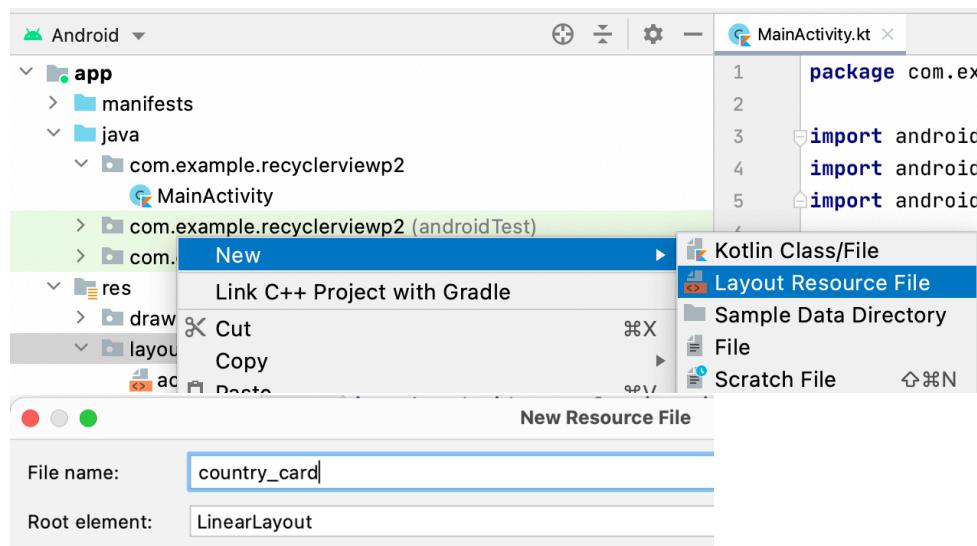


```

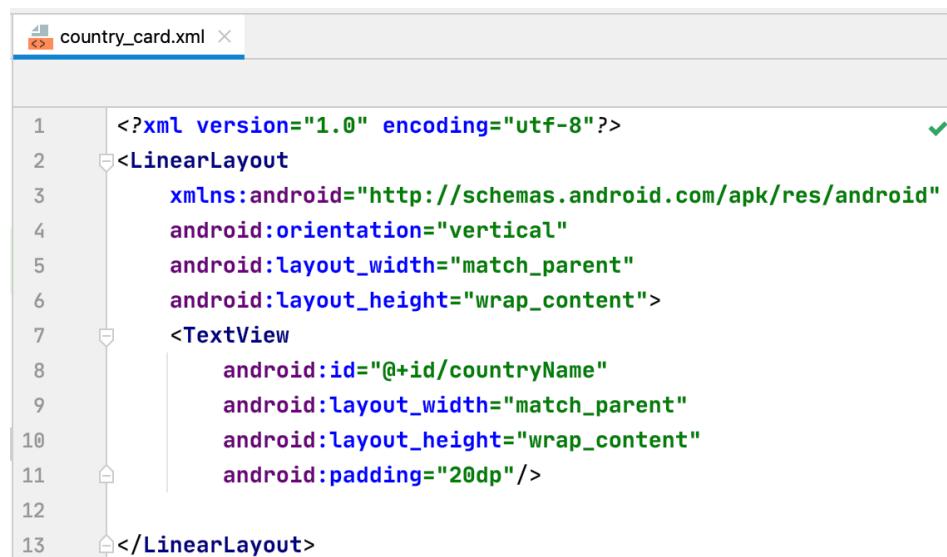
1 package com.example.recyclerviewp2
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import androidx.recyclerview.widget.RecyclerView
6
7 class MainActivity : AppCompatActivity() {
8     //zmienna obsługująca listę dziedziczącą po RecyclerView
9     lateinit var cantryList:RecyclerView
10    override fun onCreate(savedInstanceState: Bundle?) {
11        super.onCreate(savedInstanceState)
12        setContentView(R.layout.activity_main)
13    }
14 }

```

Dodajemy layout będący pojedynczą kartą reprezentującą element listy:



Modyfikujemy kod według swoich potrzeb:

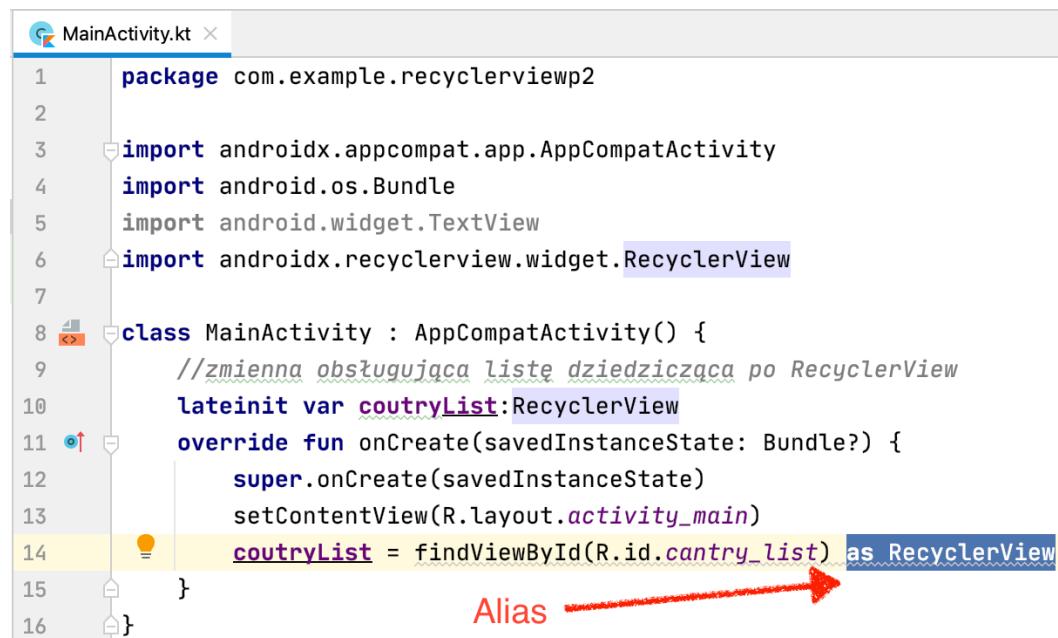


```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TextView
        android:id="@+id/countryName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="20dp"/>
</LinearLayout>

```

W pliku kotlinowym MainActivity.kt nadajemy wartość utworzonej wcześniej zmiennej:



```

package com.example.recyclerviewp2

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

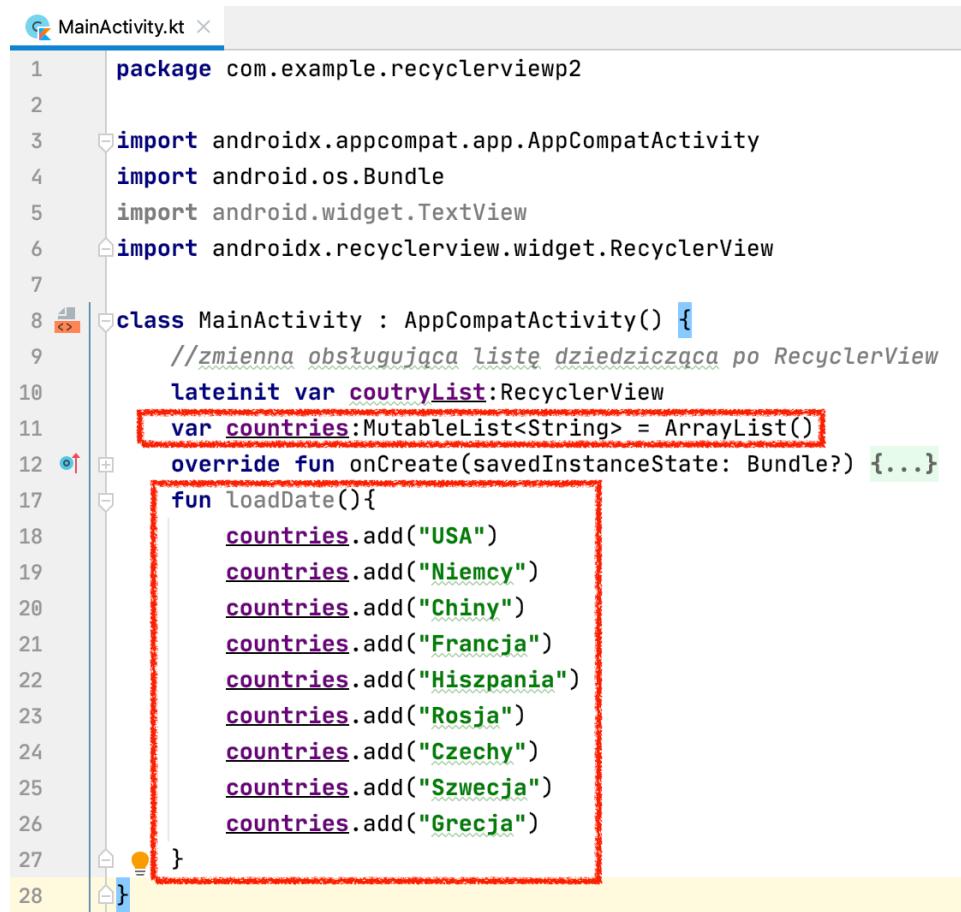
class MainActivity : AppCompatActivity() {
    //zmienna obsługująca listę dziedziczącą po RecyclerView
    lateinit var countryList: RecyclerView

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        countryList = findViewById(R.id.country_list) as RecyclerView
    }
}

```

A red arrow points from the word 'Alias' to the 'as RecyclerView' part of the code, indicating that 'countryList' is an alias for the RecyclerView object.

Dodajemy także zmienną reprezentującą tablicę z elementami do wyświetlenia:

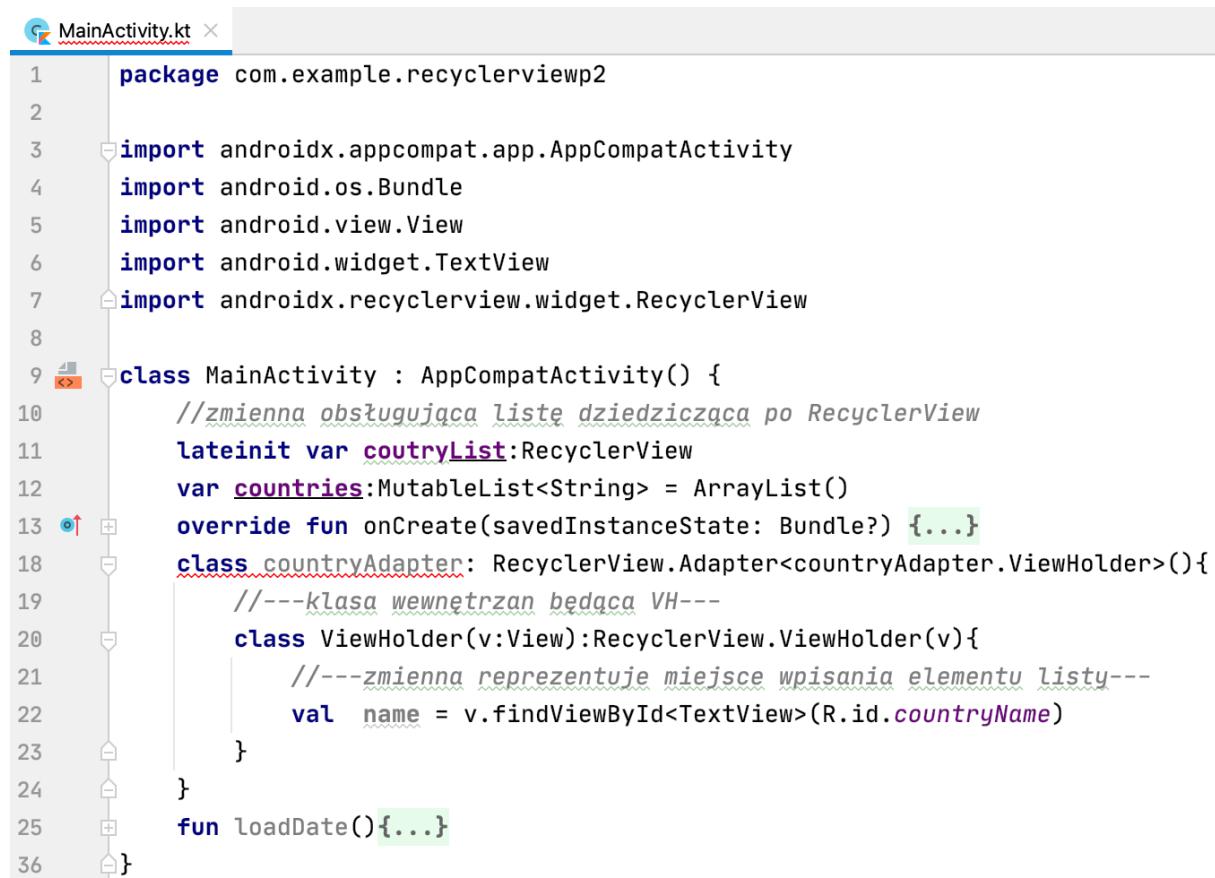


```

1 package com.example.recyclerviewp2
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.widget.TextView
6 import androidx.recyclerview.widget.RecyclerView
7
8 class MainActivity : AppCompatActivity() {
9     //zmienna obsługująca listę dziedziczącą po RecyclerView
10    lateinit var countryList:RecyclerView
11    var countries:MutableList<String> = ArrayList()
12    override fun onCreate(savedInstanceState: Bundle?) {...}
13    fun loadDate(){
14        countries.add("USA")
15        countries.add("Niemcy")
16        countries.add("Chiny")
17        countries.add("Francja")
18        countries.add("Hiszpania")
19        countries.add("Rosja")
20        countries.add("Czechy")
21        countries.add("Szwecja")
22        countries.add("Grecja")
23    }
24
25
26
27
28 }

```

Tworzymy klasę adaptera tablicy „countries” do listy „loadDate”:



```

1 package com.example.recyclerviewp2
2
3 import androidx.appcompat.app.AppCompatActivity
4 import android.os.Bundle
5 import android.view.View
6 import android.widget.TextView
7 import androidx.recyclerview.widget.RecyclerView
8
9 class MainActivity : AppCompatActivity() {
10     //zmienna obsługująca listę dziedziczącą po RecyclerView
11    lateinit var countryList:RecyclerView
12    var countries:MutableList<String> = ArrayList()
13    override fun onCreate(savedInstanceState: Bundle?) {...}
14    class countryAdapter: RecyclerView.Adapter<countryAdapter.ViewHolder>{
15        //---klasa wewnętrzna będąca VH---
16        class ViewHolder(v:View):RecyclerView.ViewHolder(v){
17            //---zmienna reprezentująca miejsce wpisania elementu listy---
18            val name = v.findViewById<TextView>(R.id.countryName)
19        }
20
21        ...
22    }
23
24    fun loadDate(){...}
25
26 }

```

Teraz trzeba zaimplementować metody wyświetlania elementów listy:

The screenshot shows the Android Studio IDE. On the left, the code editor displays `MainActivity.kt` with several method signatures underlined in red, indicating they have not been implemented. On the right, the JavaDoc for `androidx.recyclerview.widget.RecyclerView.Adapter` is open, showing three abstract methods: `onCreateViewHolder`, `onBindViewHolder`, and `getItemCount`. The code editor has a light gray background, while the JavaDoc panel has a blue header.

```
13     override fun onCreate(saved
14
15     class countryAdapter: Recyc
16
17     Implement members
18
19     Make 'countryAdapter' 'abstract'
20

MainActivity.kt x
14     override fun onCreatesavedInstanceState: Bundle? {...}
15
16     class countryAdapter: RecyclerView.Adapter<countryAdapter.ViewHolder>(){
17
18         override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {...}
19
20         override fun onBindViewHolder(holder: ViewHolder, position: Int) {...}
21
22         override fun getItemCount(): Int {...}
23
24
25         //---klaa wejnetrzan bedaca VH---
26         class ViewHolder(v:View):RecyclerView.ViewHolder(v){
```

Klasa powinna dostać dwa argumenty: listę i kontekst:

```
15     override fun onActivityCreated(savedInstanceState: Bundle?) { ... }
16
17     class countryAdapter(items:List<String>,ctx:Context): RecyclerView
18         var lista = items
19         var context = ctx
20
21     override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): CountryHolder {
22         val view = LayoutInflater.from(context).inflate(R.layout.item_country, parent, false)
23         return CountryHolder(view)
24     }
25
26     override fun onBindViewHolder(holder: CountryHolder, position: Int) {
27         holder.bind(lista[position])
28     }
29
30     override fun getItemCount(): Int = lista.size
31 }
```

Metoda zliczajaca ilosc elementow listy i zwracajaca liczbe do aktywnosci:

```
31    override fun getItemCount(): Int {  
32        return lista.size  
33    }  
  
24    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): Viewholder {  
25        //---uchwyt do szablonu listy---  
26        return Viewholder(LayoutInflater.from(context).inflate(R.layout.country_card, parent, attachToRoot: false))  
27    }  
  
37    override fun onBindViewHolder(holder: Viewholder, position: Int) {  
38        holder.name.text = lista.get(position)  
39    }
```

Teraz wystarczy załadować dane oraz wywołać obiekty typu layout i adapter w odniesieniu do listy:

```
MainActivity.kt x
17
18     var countries:MutableList<String> = ArrayList()
19     override fun onCreate(savedInstanceState: Bundle?) {
20         super.onCreate(savedInstanceState)
21         setContentView(R.layout.activity_main)
22         loadDate()
23         countrylist = findViewById(R.id.cantry_list).as RecyclerView
24         //---sposoby wyświetlania listy liniowy lub gridowy---
25         //countryList.layoutManager = LinearLayoutManager(this)
26         countryList.layoutManager = GridLayoutManager(context: this, spanCount: 2)
27         countryList.adapter = CountryAdapter(countries, ctx: this)
28     }
29 }
```