###### МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

###### КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

###### Институт фундаментальных наук

**Кафедра ЮНЕСКО по новым информационным технологиям**

ОТЧЕТ

О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ

«РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

ЛАБОРАТОРНАЯ РАБОТА №8»

студента 2 курса, ПИ-202 группы

**Садовикова Никиты Евгеньевича**

Направления подготовки:

09.03.03 – «Прикладная информатика (прикладная информатика в экономике)»

Руководитель:

доцент, кандидат технических наук

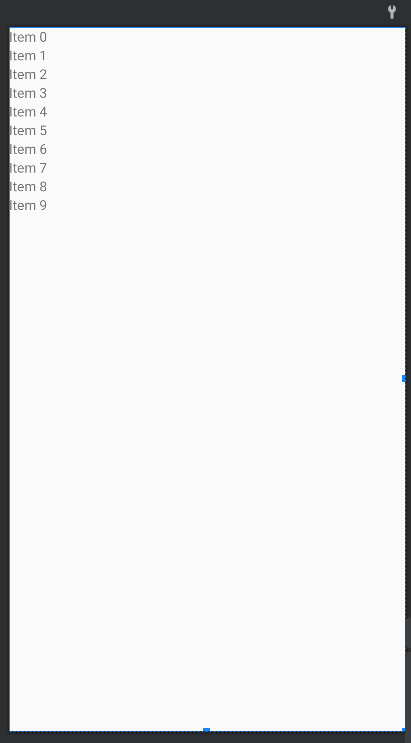
С.Ю. Завозкин

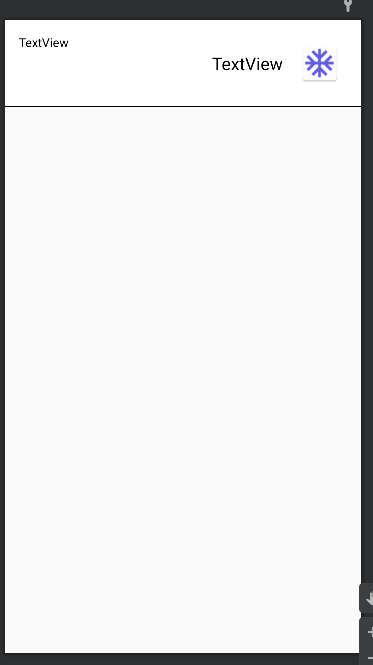
Работа защищена

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_»

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Кемерово 2021

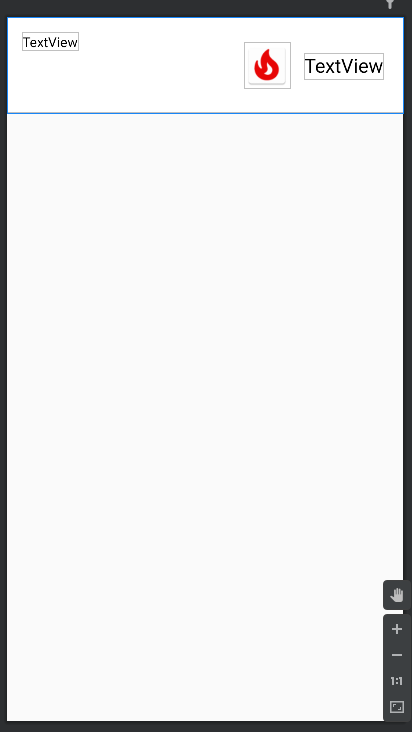
Расставляем элементы в макет и создаём адаптер с MainActivity отдельными классами  




Элемент View, который выводится тогда, когда погода отрицательная.

XML код лайаута

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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:id="@+id/linearLayout2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:background="#FFFFFF">  
  
 <TextView  
 android:id="@+id/temperature\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="15dp"  
 android:text="TextView"  
 android:textColor="#000000"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/imageView"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/datetime\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textColor="#000000"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.045"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.197" />  
  
 <View  
 android:id="@+id/view1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="1sp"  
 android:background="@android:color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="1.0" />  
  
 <ImageView  
 android:id="@+id/imageView"  
 android:layout\_width="55dp"  
 android:layout\_height="50dp"  
 android:layout\_marginEnd="20dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/ic\_launcher" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

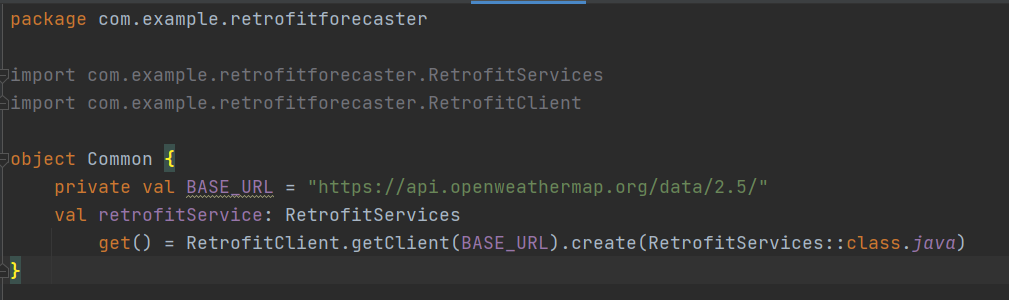


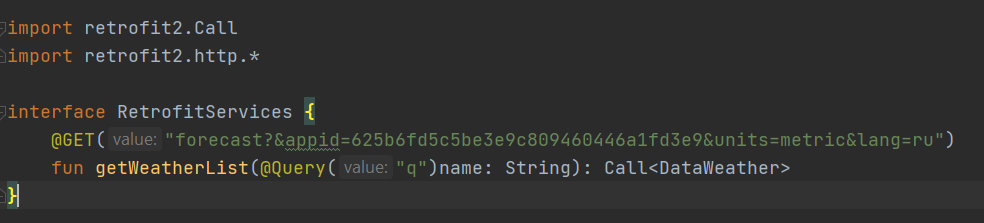
Элемент View, который выводится тогда, когда погода положительная.

XML код этого лайаута

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.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:id="@+id/linearLayout2"  
 android:layout\_width="match\_parent"  
 android:layout\_height="100dp"  
 android:background="#FFFFFF">  
  
  
  
 <TextView  
 android:id="@+id/temperature\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="20dp"  
 android:text="TextView"  
 android:textColor="#000000"  
 android:textSize="20sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
 <TextView  
 android:id="@+id/datetime\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="TextView"  
 android:textColor="#000000"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.045"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.197" />  
  
 <View  
 android:id="@+id/view1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="1sp"  
 android:background="@android:color/black"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="1.0" />  
  
 <ImageView  
 android:id="@+id/imageView2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="15dp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toStartOf="@+id/temperature\_text"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:srcCompat="@drawable/fire" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>

Файлы с ссылкой API



Функция получения погоды из города краснодар с передачей листа в адаптер

import android.os.Bundle  
import android.util.Log  
import androidx.appcompat.app.AppCompatActivity  
import androidx.recyclerview.widget.LinearLayoutManager  
import androidx.recyclerview.widget.RecyclerView  
import retrofit2.Call  
import retrofit2.Callback  
import retrofit2.Response  
  
  
class MainActivity : AppCompatActivity() {  
 lateinit var mService: RetrofitServices  
 lateinit var layoutManager: LinearLayoutManager  
 val adapter = Adapter()  
 var listweather = *listOf*<ListItem>()  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
 val recyclerView: RecyclerView = findViewById(R.id.*rView*)  
 *title* = "Krasnodar"  
 mService = Common.retrofitService  
 layoutManager = LinearLayoutManager(this)  
 recyclerView.*layoutManager* = layoutManager  
 recyclerView.*adapter* = adapter  
  
 getAllWeatherList("Krasnodar")  
 }  
  
 fun getAllWeatherList(city: String) {  
 mService.getWeatherList(city).enqueue(object : Callback<DataWeather> {  
 override fun onResponse(call: Call<DataWeather>, response: Response<DataWeather>) {  
 val abc = response.body() as DataWeather  
 listweather = abc.list  
 Log.d("123", abc.cod)  
 adapter.submitList(listweather)  
 adapter.notifyDataSetChanged()  
 }  
 override fun onFailure(call: Call<DataWeather>, t: Throwable) {  
 Log.d("123", t.message.*toString*())  
 }  
 })  
 }  
}

Класс адаптера с DiffUtil и несколькими ViewHolder’ами

import android.graphics.Color  
import android.view.LayoutInflater  
import android.view.View  
import android.view.ViewGroup  
import android.widget.TextView  
import androidx.recyclerview.widget.DiffUtil  
import androidx.recyclerview.widget.ListAdapter  
import androidx.recyclerview.widget.RecyclerView  
  
class ContactItemDiffCallback : DiffUtil.ItemCallback<ListItem>(){  
 override fun areItemsTheSame(oldItem: ListItem, newItem: ListItem) = oldItem == newItem  
  
 override fun areContentsTheSame(oldItem: ListItem, newItem: ListItem) = oldItem == newItem  
  
}  
  
class ViewHolderCold(view: View) : RecyclerView.ViewHolder(view) {  
 val datetime: TextView = view.findViewById(R.id.*datetime\_text*)  
 val temperature: TextView = view.findViewById(R.id.*temperature\_text*)  
  
  
 fun bindTo(weather: ListItem, position: Int){  
 if (temperature != null) {temperature.*text* = weather.main.temp.toString()}  
 if (datetime != null) {datetime.*text* = weather.dt\_txt}  
  
 }  
}  
  
class ViewHolderHot(view: View) : RecyclerView.ViewHolder(view) {  
 val datetime: TextView = view.findViewById(R.id.*datetime\_text*)  
 val temperature: TextView = view.findViewById(R.id.*temperature\_text*)  
  
  
  
 fun bindTo(weather: ListItem){  
 if (temperature != null) {temperature.*text* = weather.main.temp.toString()}  
 if (datetime != null) {datetime.*text* = weather.dt\_txt}  
 }  
}  
  
class Adapter : ListAdapter<ListItem, RecyclerView.ViewHolder>(ContactItemDiffCallback()){  
  
 override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {  
 if(viewType == 0){ val view = LayoutInflater.from(parent.*context*).inflate(R.layout.*rview\_item*, parent, false)  
 view.setBackgroundColor(Color.rgb(208, 65, 65))  
 return ViewHolderHot(view)}  
 else{  
 val view = LayoutInflater.from(parent.*context*).inflate(R.layout.*rview\_item2*, parent, false)  
 view.setBackgroundColor(Color.rgb(26, 103, 212))  
 return ViewHolderCold(view)  
 }  
 }  
 override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {  
 val data = *currentList*[position]  
 if(data.main.temp>0) {  
 val viewholderhot: ViewHolderHot = holder as ViewHolderHot  
 viewholderhot.bindTo(data)  
 }else{  
 val viewholdercold: ViewHolderCold = holder as ViewHolderCold  
 viewholdercold.bindTo(data,position)  
 }  
 }  
 override fun getItemViewType(position: Int): Int {  
 return if (*currentList*[position].main.temp>0) {  
 0  
 } else {  
 1  
 }  
 }  
}

Итоговое приложение

