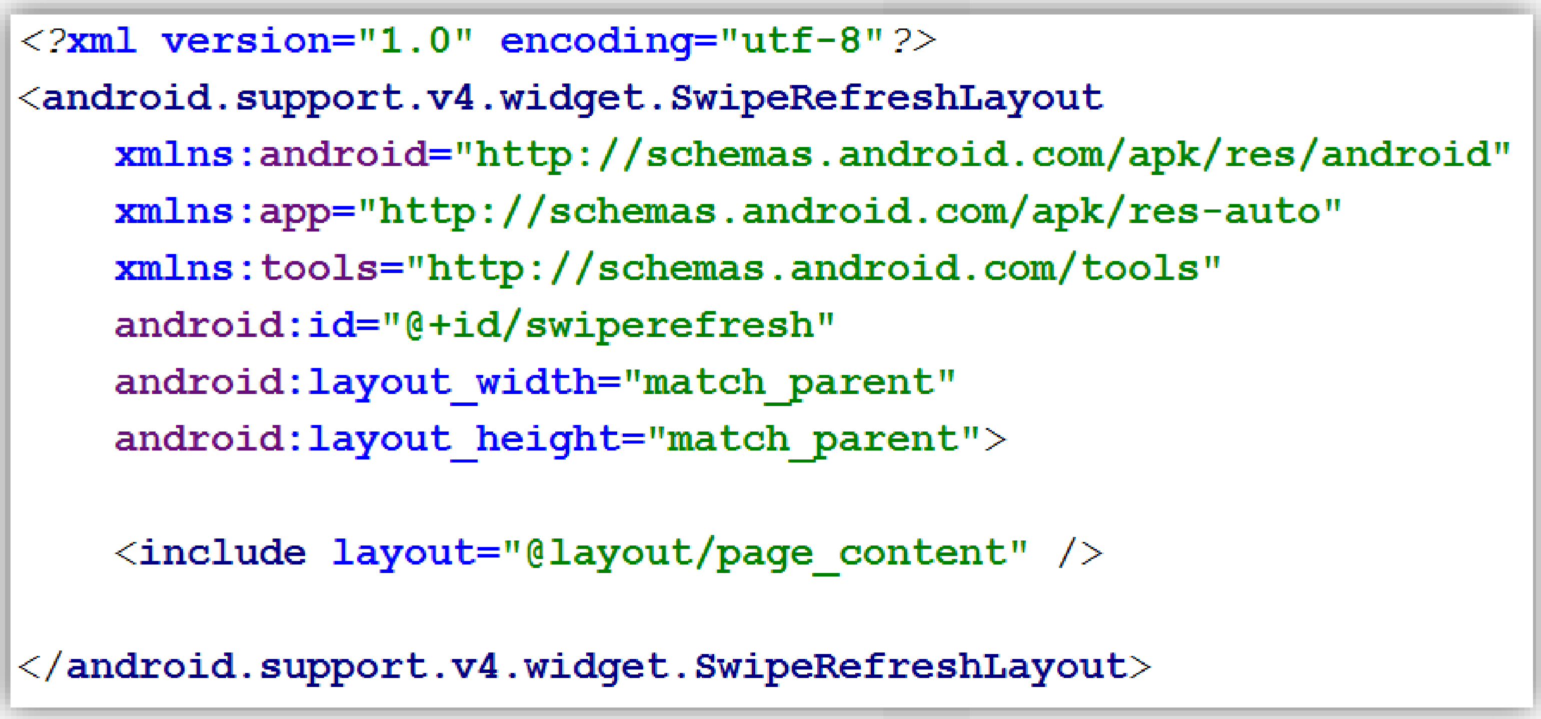
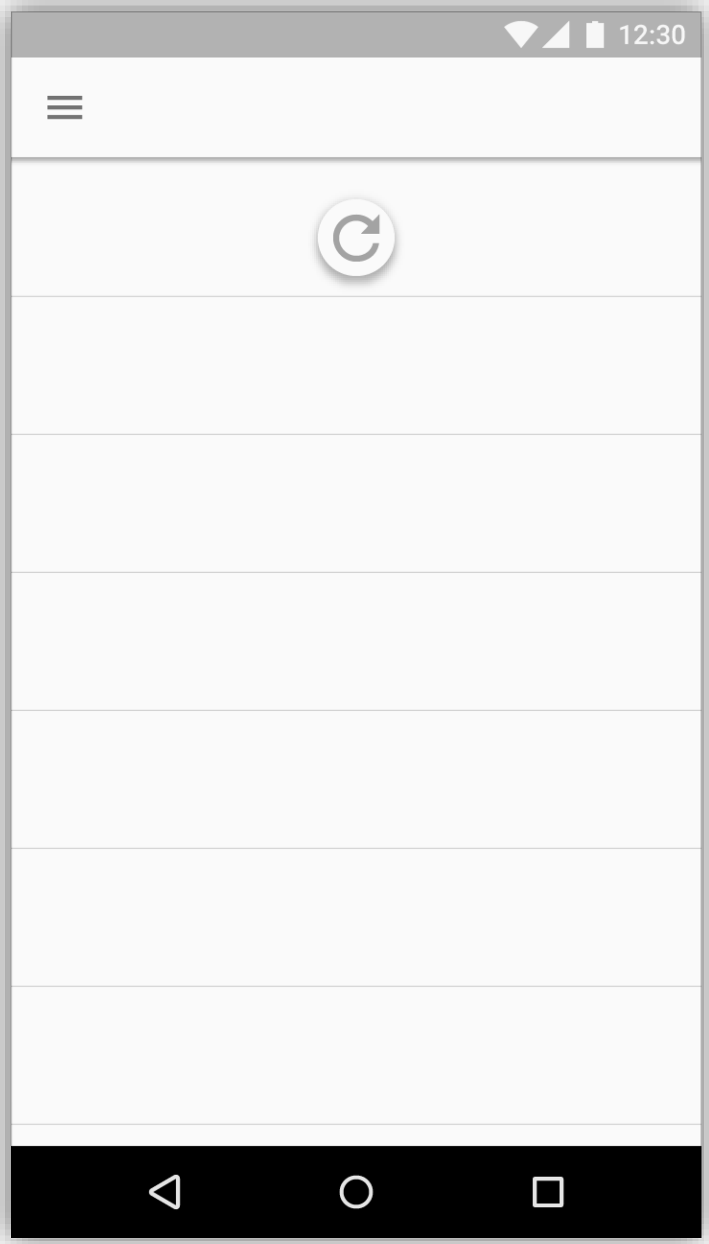
# Komunikacja HTTP (II)

Marek Wojtuszkiewicz

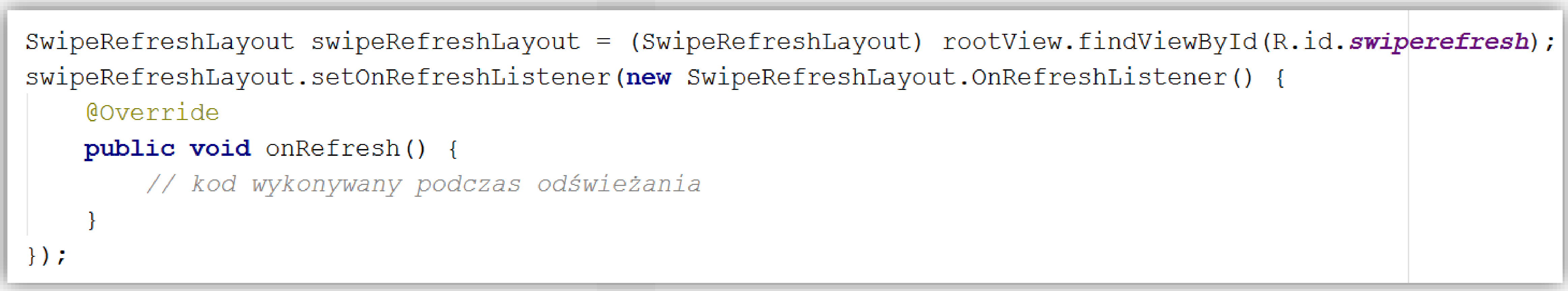
# SwipeRefreshLayout



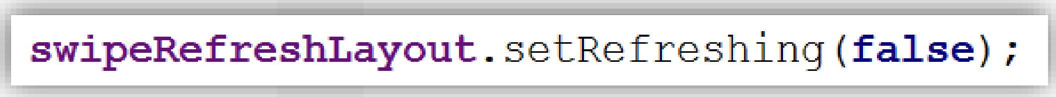
```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v4.widget.SwipeRefreshLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/swiperefresh"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <include layout="@layout/page_content" />

</android.support.v4.widget.SwipeRefreshLayout>
```



# SwipeRefreshLayout



```java
SwipeRefreshLayout swipeRefreshLayout = (SwipeRefreshLayout) rootView.findViewById(R.id.swiperefresh);
swipeRefreshLayout.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
    @Override
    public void onRefresh() {
        // kod wykonywany podczas odświeżania
    }
});
```



```java
swipeRefreshLayout.setRefreshing(false);
```

# RecyclerView

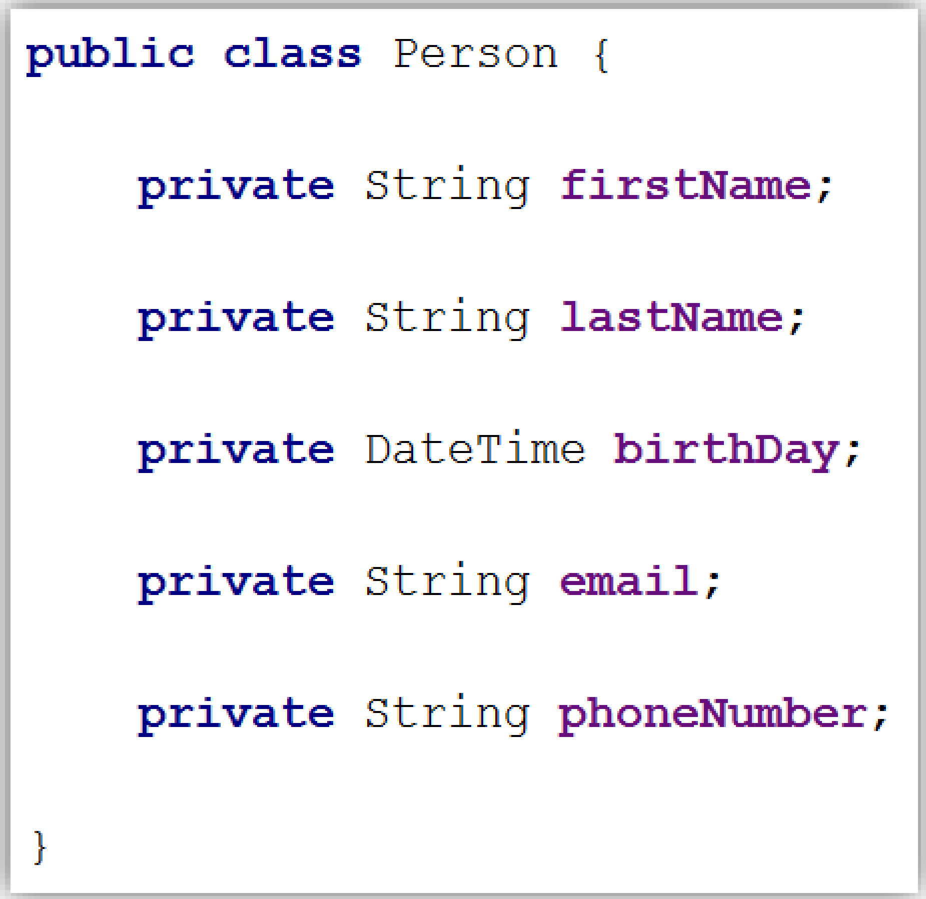
Bardziej zaawansowany i elastyczny od ListView

Layout managers (LinearLayoutManager, GridLayoutManager, StaggeredGridLayoutManager)

Animacje elementów

# RecyclerView (1/5)

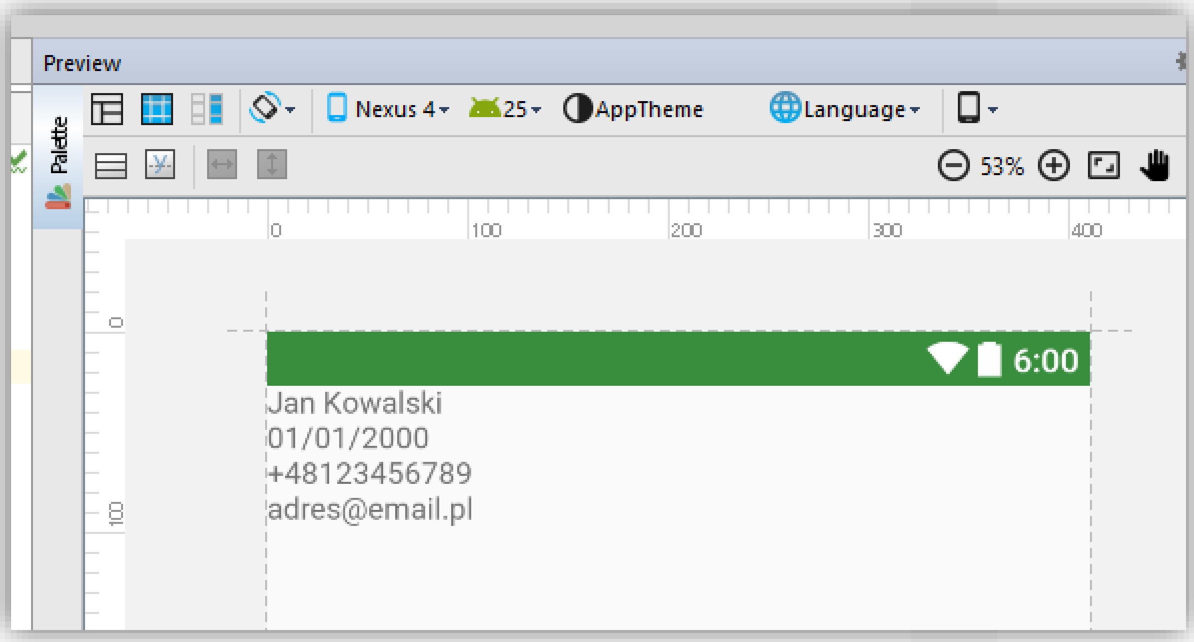
Model danych



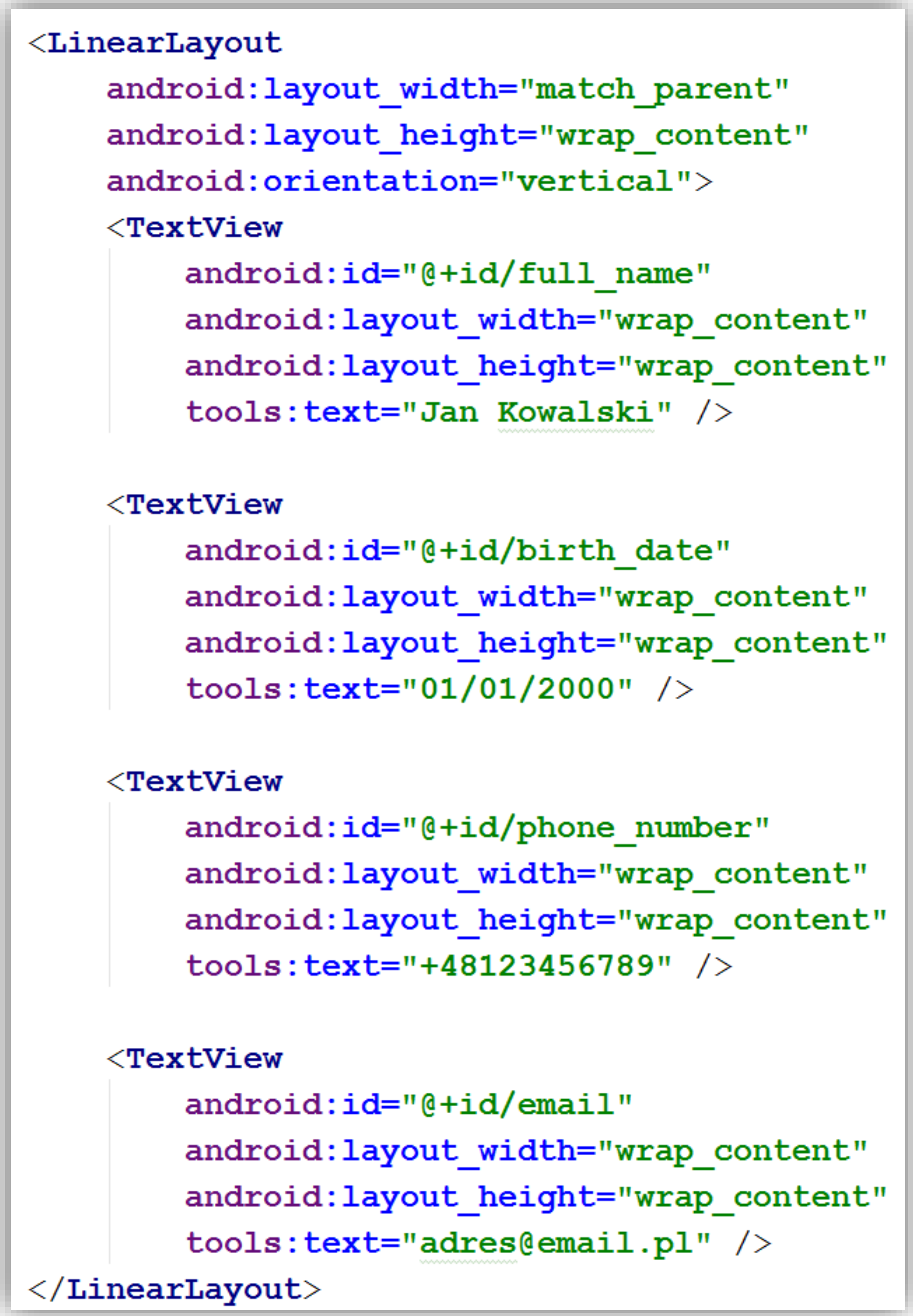
```
public class Person {

    private String firstName;

    private String lastName;

    private DateTime birthDay;

    private String email;

    private String phoneNumber;

}
```

# RecyclerView (2/5)

## Widok danych



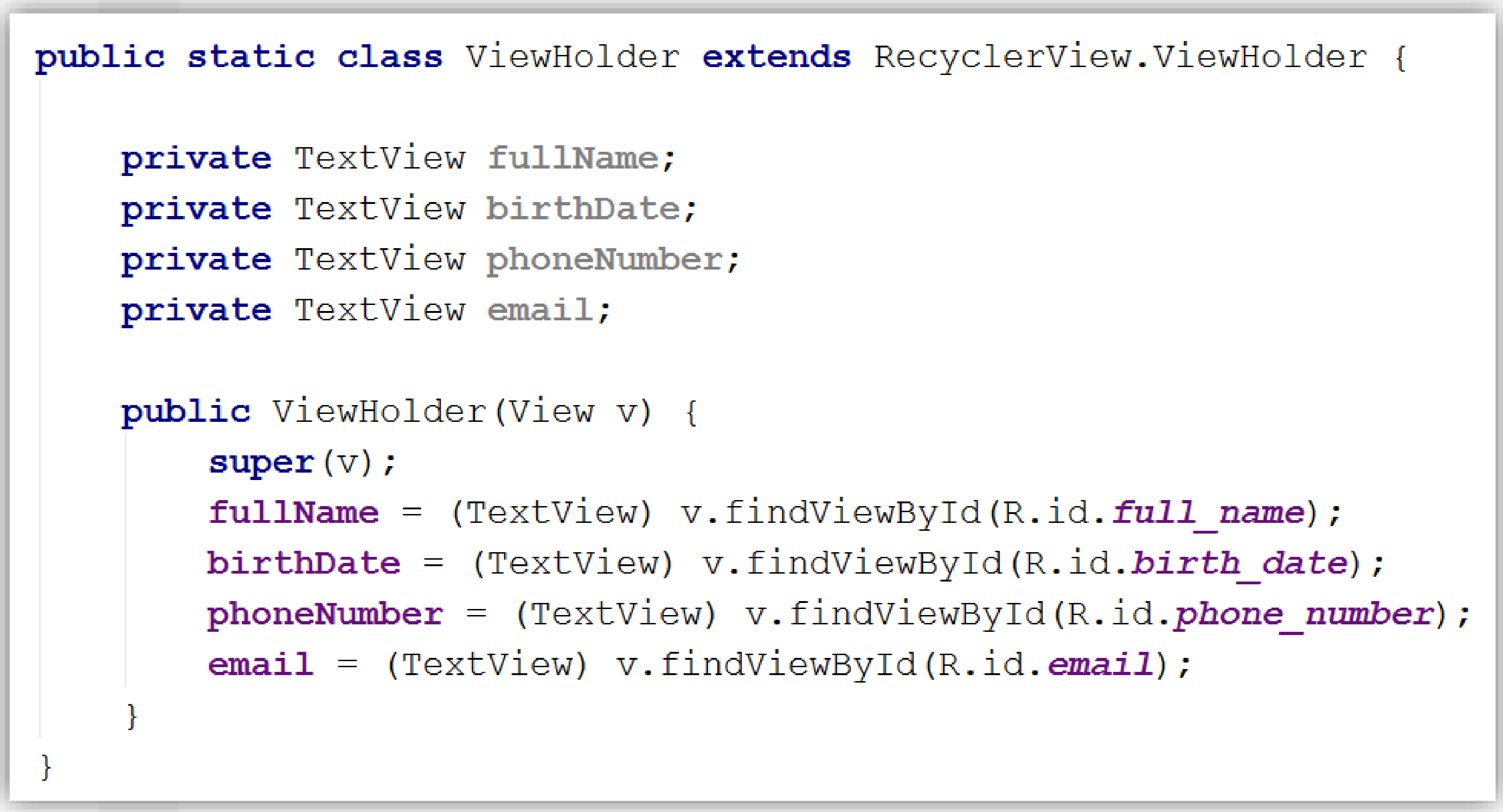
Jan Kowalski
01/01/2000
+48123456789
adres@email.pl



```xml
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
    <TextView
        android:id="@+id/full_name"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="Jan Kowalski" />

    <TextView
        android:id="@+id/birth_date"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="01/01/2000" />

    <TextView
        android:id="@+id/phone_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="+48123456789" />

    <TextView
        android:id="@+id/email"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        tools:text="adres@email.pl" />
</LinearLayout>
```

# RecyclerView (3/5)

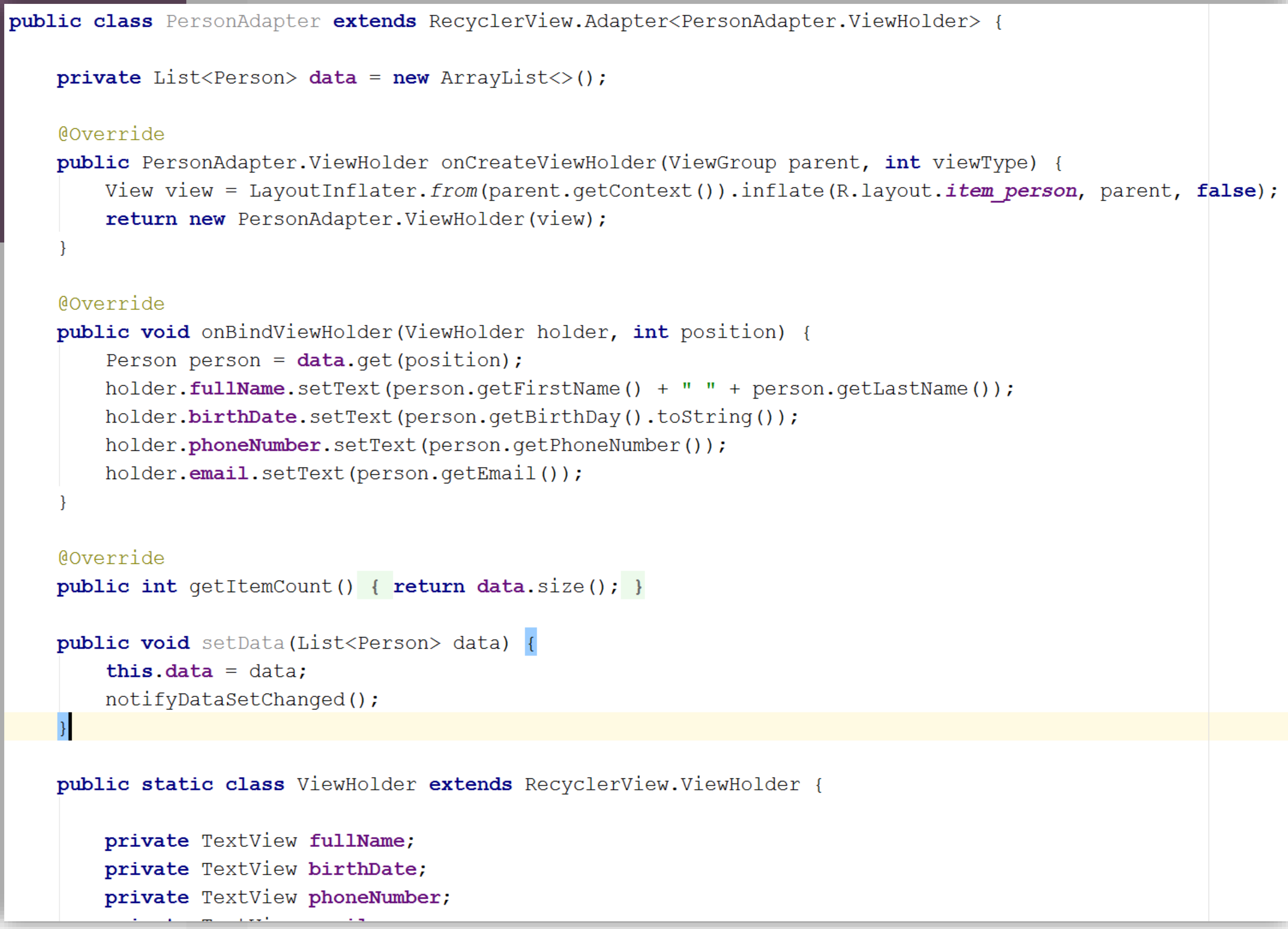
RecyclerView.ViewHolder



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

    private TextView fullName;
    private TextView birthDate;
    private TextView phoneNumber;
    private TextView email;

    public ViewHolder(View v) {
        super(v);
        fullName = (TextView) v.findViewById(R.id.full_name);
        birthDate = (TextView) v.findViewById(R.id.birth_date);
        phoneNumber = (TextView) v.findViewById(R.id.phone_number);
        email = (TextView) v.findViewById(R.id.email);
    }
}
```

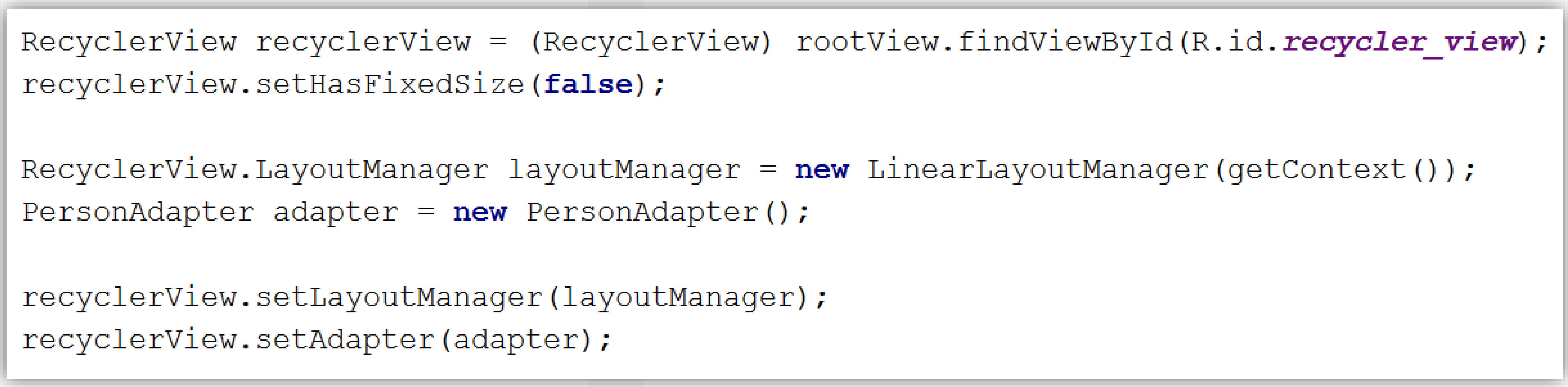
# RecyclerView (4/5)

RecyclerView.Adapter



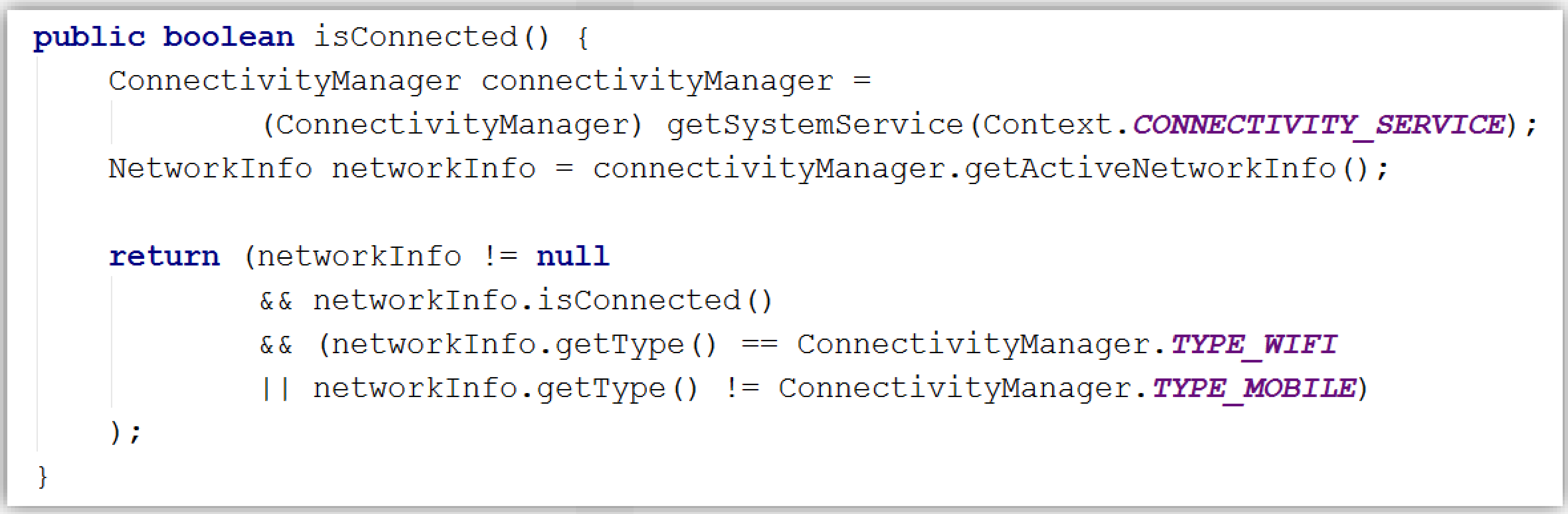
```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder> {

    private List<Person> data = new ArrayList<>();

    @Override
    public PersonAdapter.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.item_person, parent, false);
        return new PersonAdapter.ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(ViewHolder holder, int position) {
        Person person = data.get(position);
        holder.fullName.setText(person.getFirstName() + " " + person.getLastName());
        holder.birthDate.setText(person.getBirthDay().toString());
        holder.phoneNumber.setText(person.getPhoneNumber());
        holder.email.setText(person.getEmail());
    }

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

    public void setData(List<Person> data) {
        this.data = data;
        notifyDataSetChanged();
    }

    public static class ViewHolder extends RecyclerView.ViewHolder {

        private TextView fullName;
        private TextView birthDate;
        private TextView phoneNumber;
```

# RecyclerView (5/5)



```
RecyclerView recyclerView = (RecyclerView) rootView.findViewById(R.id.recycler_view);
recyclerView.setHasFixedSize(false);

RecyclerView.LayoutManager layoutManager = new LinearLayoutManager(getContext());
PersonAdapter adapter = new PersonAdapter();

recyclerView.setLayoutManager(layoutManager);
recyclerView.setAdapter(adapter);
```

# Sprawdzenie stanu połączenia

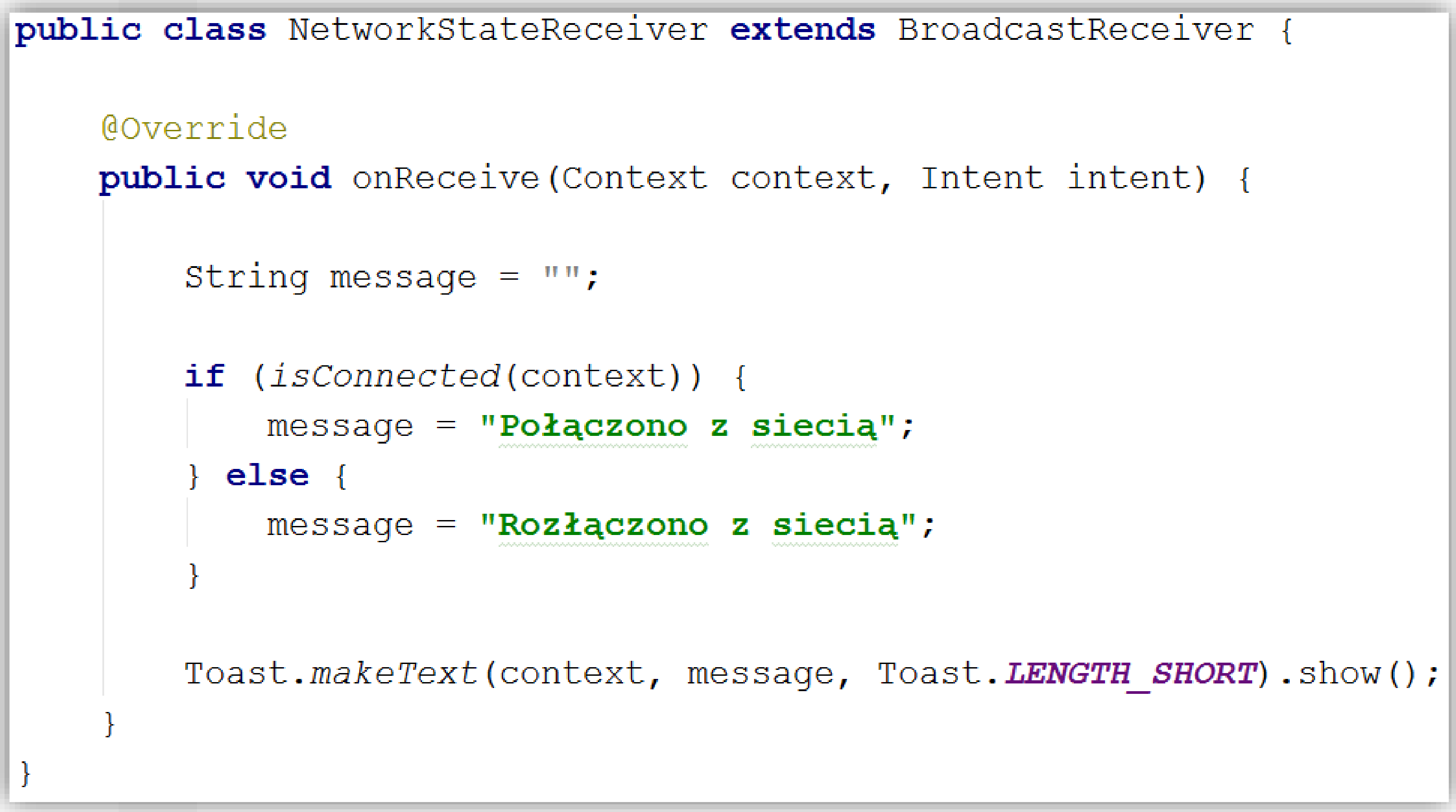


```
public boolean isConnected() {
    ConnectivityManager connectivityManager =
            (ConnectivityManager) getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connectivityManager.getActiveNetworkInfo();

    return (networkInfo != null
            && networkInfo.isConnected()
            && (networkInfo.getType() == ConnectivityManager.TYPE_WIFI
            || networkInfo.getType() != ConnectivityManager.TYPE_MOBILE)
    );
}
```

# BroadcastReceiver

Implementacja

(sposób 1)

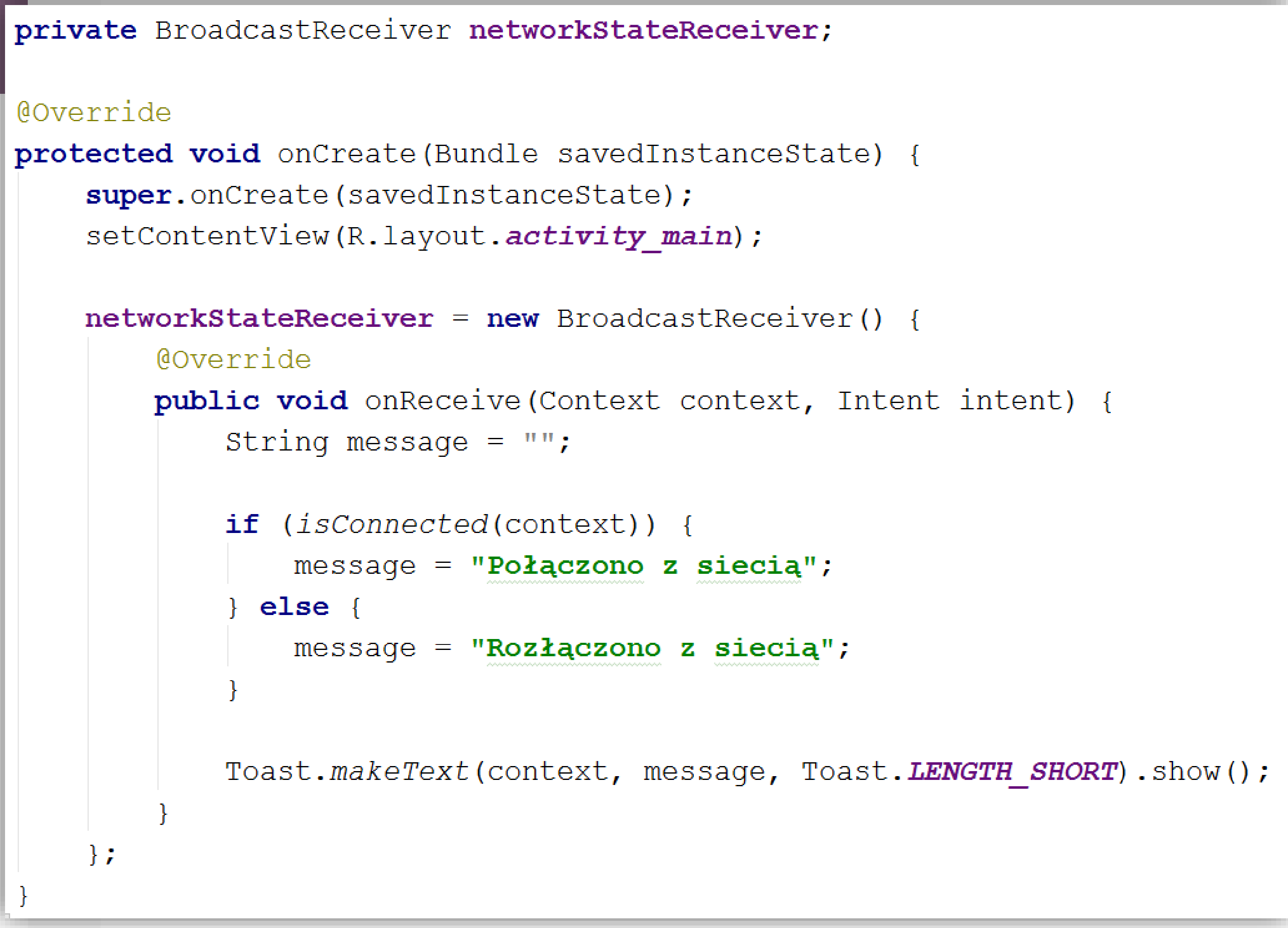


```
public class NetworkStateReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {

        String message = "";

        if (isConnected(context)) {
            message = "Połączono z siecią";
        } else {
            message = "Rozłączono z siecią";
        }

        Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
    }
}
```

# BroadcastReceiver

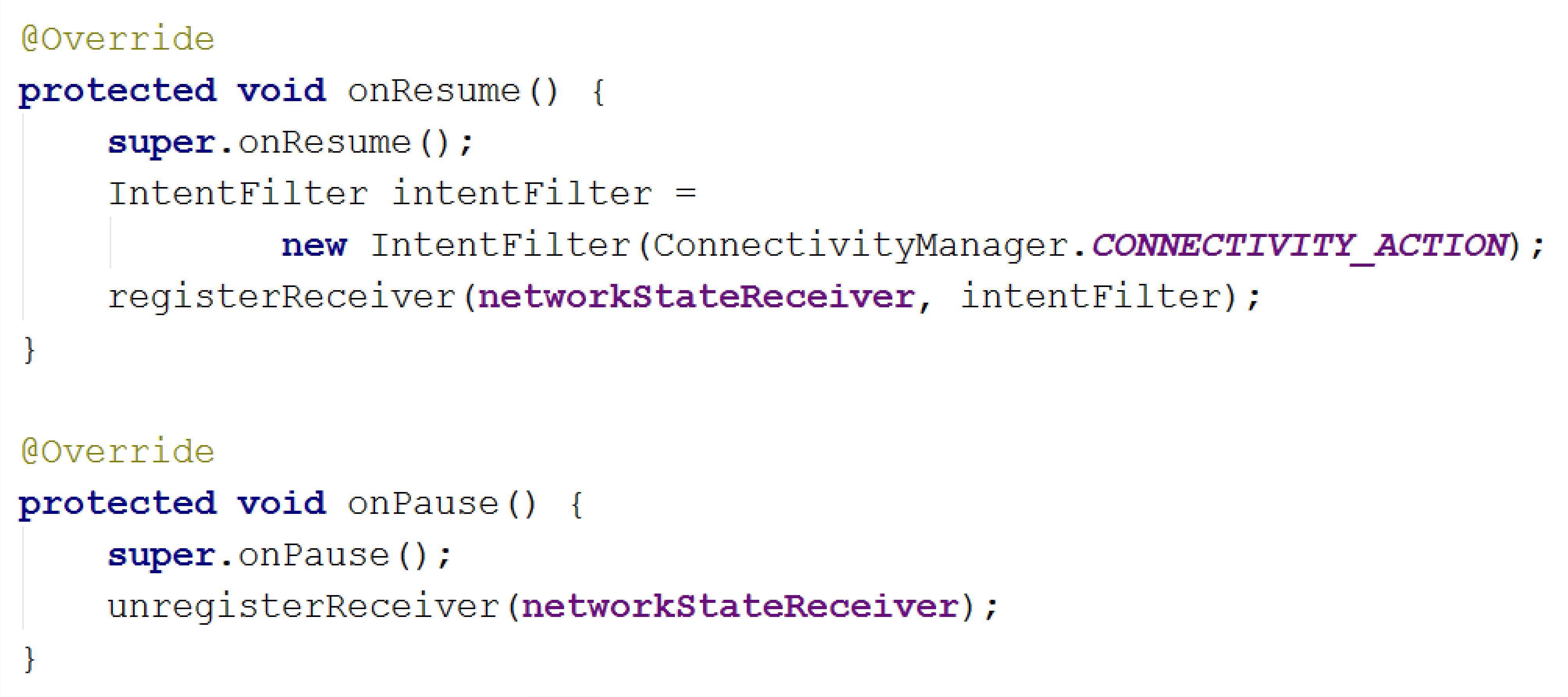
Implementacja

(sposób 2)



```java
private BroadcastReceiver networkStateReceiver;

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

    networkStateReceiver = new BroadcastReceiver() {
        @Override
        public void onReceive(Context context, Intent intent) {
            String message = "";

            if (isConnected(context)) {
                message = "Połączono z siecią";
            } else {
                message = "Rozłączono z siecią";
            }

            Toast.makeText(context, message, Toast.LENGTH_SHORT).show();
        }
    };
}
```

# Rejestracja BroadcastReceivera



```java
@Override
protected void onResume() {
    super.onResume();
    IntentFilter intentFilter =
            new IntentFilter(ConnectivityManager.CONNECTIVITY_ACTION);
    registerReceiver(networkStateReceiver, intentFilter);
}

@Override
protected void onPause() {
    super.onPause();
    unregisterReceiver(networkStateReceiver);
}
```

# Komunikacja HTTP (II) - zadanie

Rozszerz aplikację służącą wyświetlaniu informacji o aktualnej pogodzie w wybranym przez Ciebie miejscu.

1. Wyświetl w aplikacji 5- lub 16-dniową prognozę pogody
2. Kolekcję prognoz dla wybranych dni/godzin wyświetl w RececylerView
3. Dane powinny być odświeżane za pomocą „swipe to refresh"
4. Użytkownik powinien być poinformowany o niedostępności sieci

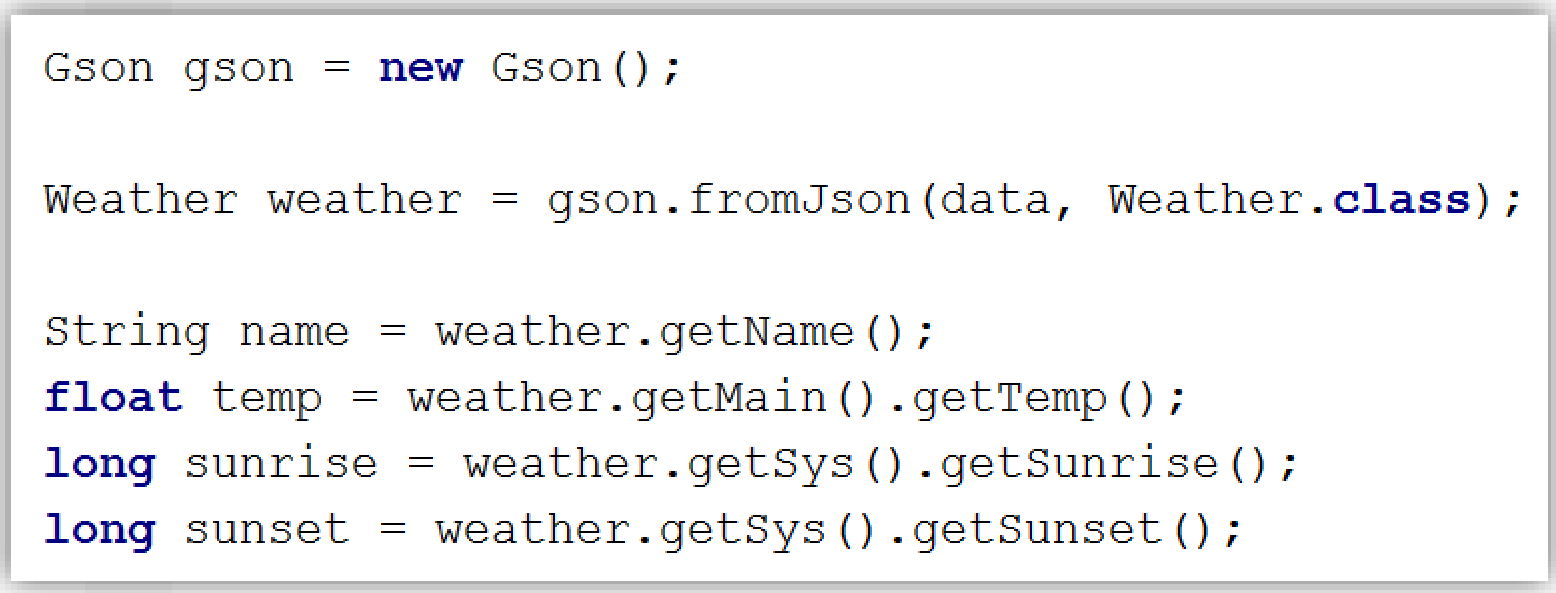
Wykorzystaj jedno z poniższych API:

https://openweathermap.org/forecast5
https://openweathermap.org/forecast16

# Bonus: gson

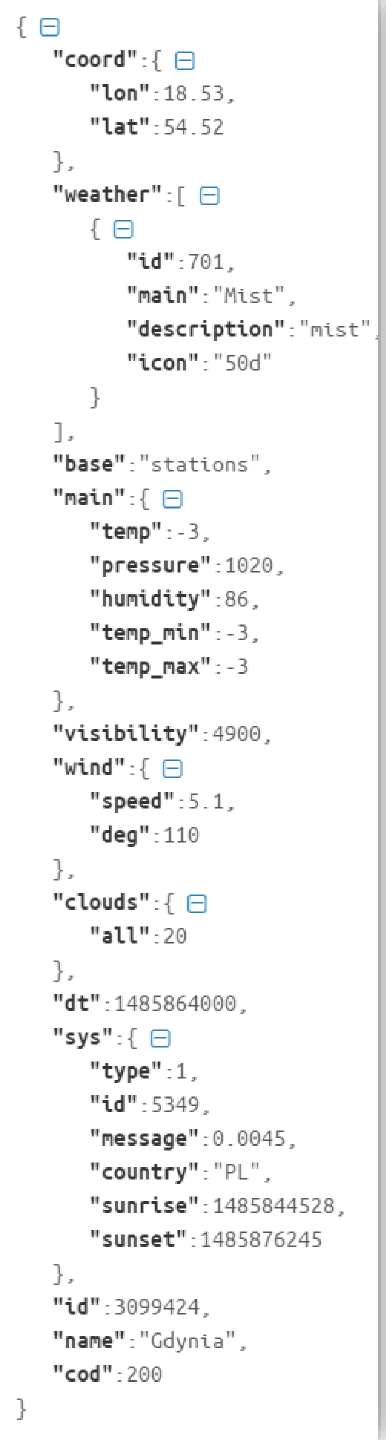
Biblioteka służąca serializacji i deserializacji obiektów języka Java do/z formatu JSON

Gradle:
com.google.code.gson:gson:2.8.0

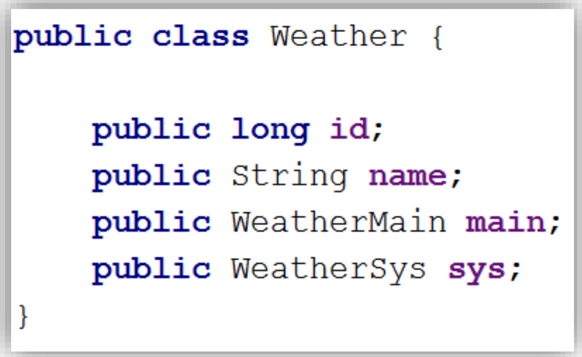


```java
Gson gson = new Gson();

Weather weather = gson.fromJson(data, Weather.class);

String name = weather.getName();
float temp = weather.getMain().getTemp();
long sunrise = weather.getSys().getSunrise();
long sunset = weather.getSys().getSunset();
```

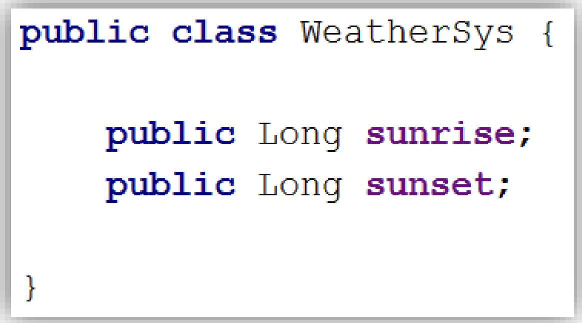
# Bonus: gson



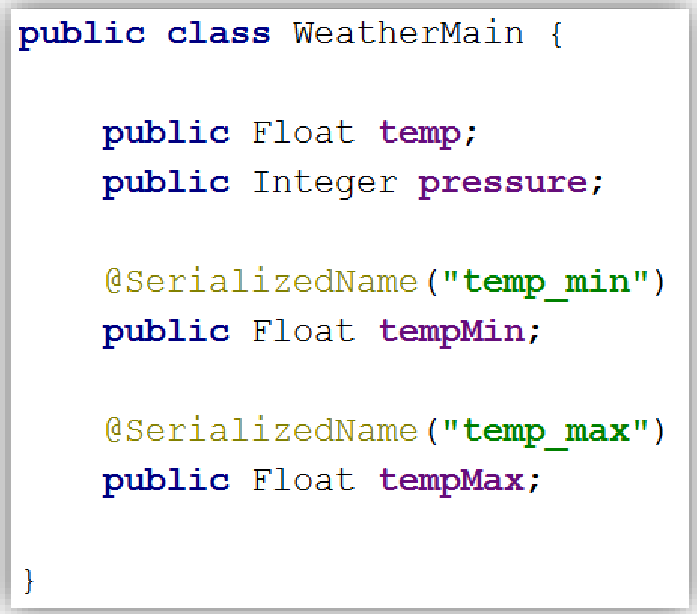
```
{ ⊟
   "coord":{ ⊟
      "lon":18.53,
      "lat":54.52
   },
   "weather":[ ⊟
      { ⊟
         "id":701,
         "main":"Mist",
         "description":"mist",
         "icon":"50d"
      }
   ],
   "base":"stations",
   "main":{ ⊟
      "temp":-3,
      "pressure":1020,
      "humidity":86,
      "temp_min":-3,
      "temp_max":-3
   },
   "visibility":4900,
   "wind":{ ⊟
      "speed":5.1,
      "deg":110
   },
   "clouds":{ ⊟
      "all":20
   },
   "dt":1485864000,
   "sys":{ ⊟
      "type":1,
      "id":5349,
      "message":0.0045,
      "country":"PL",
      "sunrise":1485844528,
      "sunset":1485876245
   },
   "id":3099424,
   "name":"Gdynia",
   "cod":200
}
```



```java
public class Weather {

    public long id;
    public String name;
    public WeatherMain main;
    public WeatherSys sys;
}
```



```java
public class WeatherSys {

    public Long sunrise;
    public Long sunset;

}
```



```java
public class WeatherMain {

    public Float temp;
    public Integer pressure;

    @SerializedName("temp_min")
    public Float tempMin;

    @SerializedName("temp_max")
    public Float tempMax;

}
```