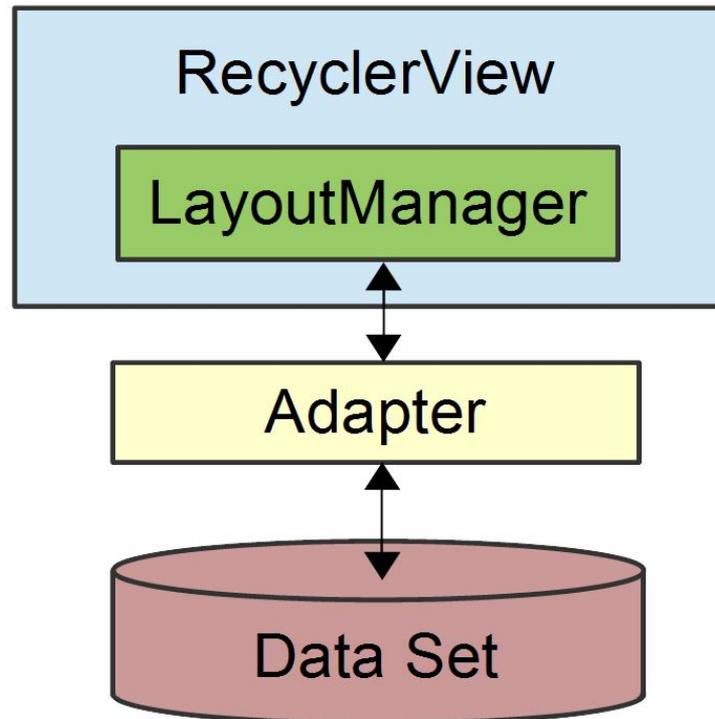


RecyclerView и CardView

<https://ziginer.github.io/RecyclerView/>

<http://developer.alexanderklimov.ru/android/views/recyclerview-kot.php>



Компонент **RecyclerView** (переиспользуемый, переработчик) появился в Android 5.0 и находится в разделе **Containers**.

Представляет собой прокручиваемый список с большими возможностями.

Схематично работу **RecyclerView** можно представить следующим образом.

На экране отображаются видимые элементы списка. Когда при прокрутке списка верхний элемент уходит за пределы экрана и становится невидимым, его содержимое очищается. При этом сам очищенный элемент помещается вниз экрана и заполняется новыми данными, иными словами переиспользуется.

RecyclerView делегирует задачи следующим вспомогательным классам:

Adapter—Расширяет макеты элементов (создает экземпляр содержимого файла макета) и привязывает данные к представлениям, которые отображаются в RecyclerView . Адаптер также сообщает о событиях щелчка элемента.

LayoutManager—Измеряет и позиционирует представления элементов в RecyclerView и управляет политикой для повторного использования представлений.

ViewHolder—Ищет и сохраняет ссылки на представление. Также помогает при обнаружении щелчков в представлении элементов.

ItemDecoration—Позволяет приложению добавлять специальные смещения рисования и макета к определенным представлениям для рисования разделителей между элементами, выделением и границами визуального группирования.

ItemAnimator—Определяет анимации, которые выполняются во время действий элемента или по мере внесения изменений в адаптер.

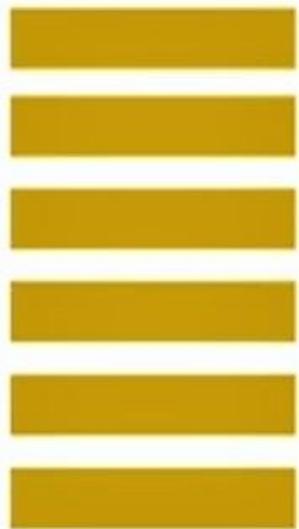
При использовании RecyclerView в приложении требуется создание производных типов следующих классов:

RecyclerView.Adapter—Предоставляет привязку из набора данных вашего приложения (относящегося к конкретному приложению) к представлениям элементов, отображаемым в RecyclerView . Адаптер знает, как связать каждую позицию представления элемента в RecyclerView определенном месте источника данных. Кроме того, адаптер обрабатывает макет содержимого в каждом представлении отдельного элемента и создает держатель представления для каждого представления. Адаптер также сообщает о событиях щелчка элемента, обнаруженных представлением элементов.

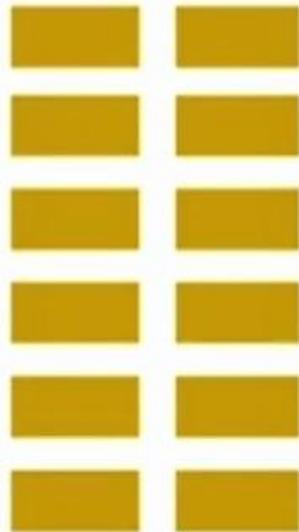
RecyclerView.ViewHolder—Кэширует ссылки на представления в файле макета элемента, чтобы поиск ресурсов не повторялся без необходимости. Владелец представления также упорядочивает события щелчка по элементам, перенаправляемые адаптеру, когда пользователь касается представления элемента, связанного с владельцем представления.

RecyclerView.LayoutManager—Позиционирует элементы внутри RecyclerView . Можно использовать один из нескольких заранее определенных диспетчеров макетов или реализовать собственный пользовательский диспетчер макетов. RecyclerView делегирует политику макета диспетчеру макетов, поэтому вы можете подключить другой диспетчер макетов без внесения значительных изменений в приложение.

Стандартные LayoutManager



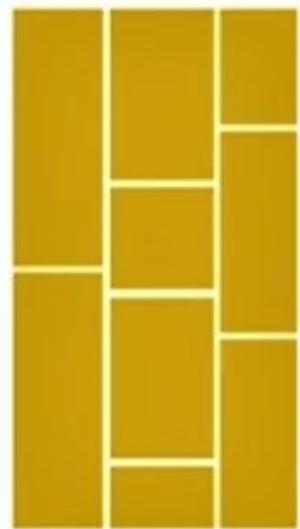
Linear



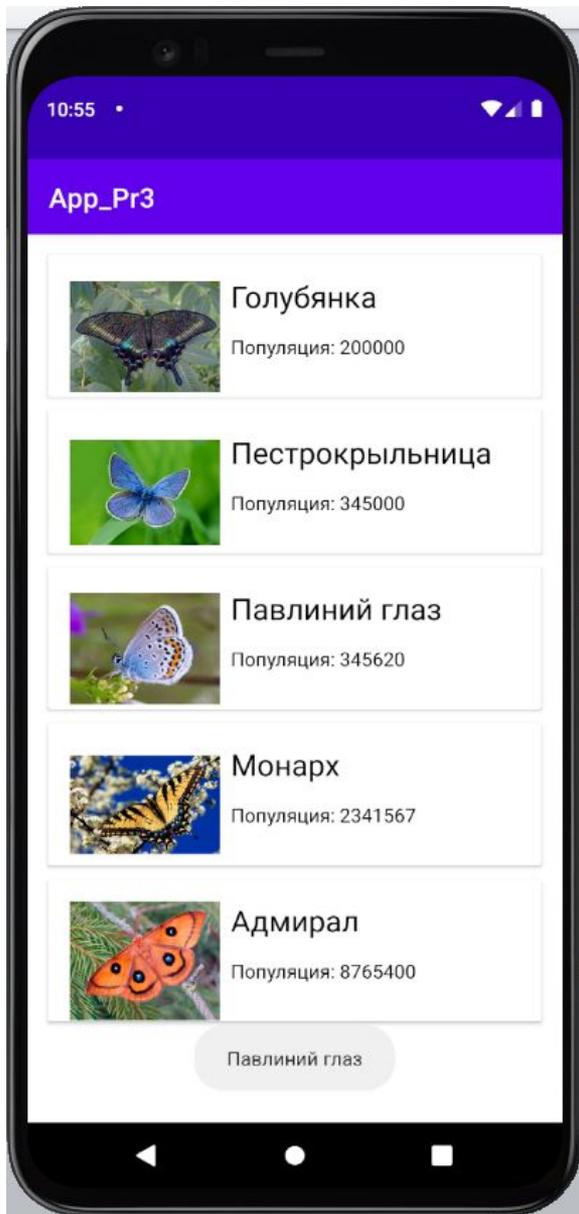
Grid



Grid + span

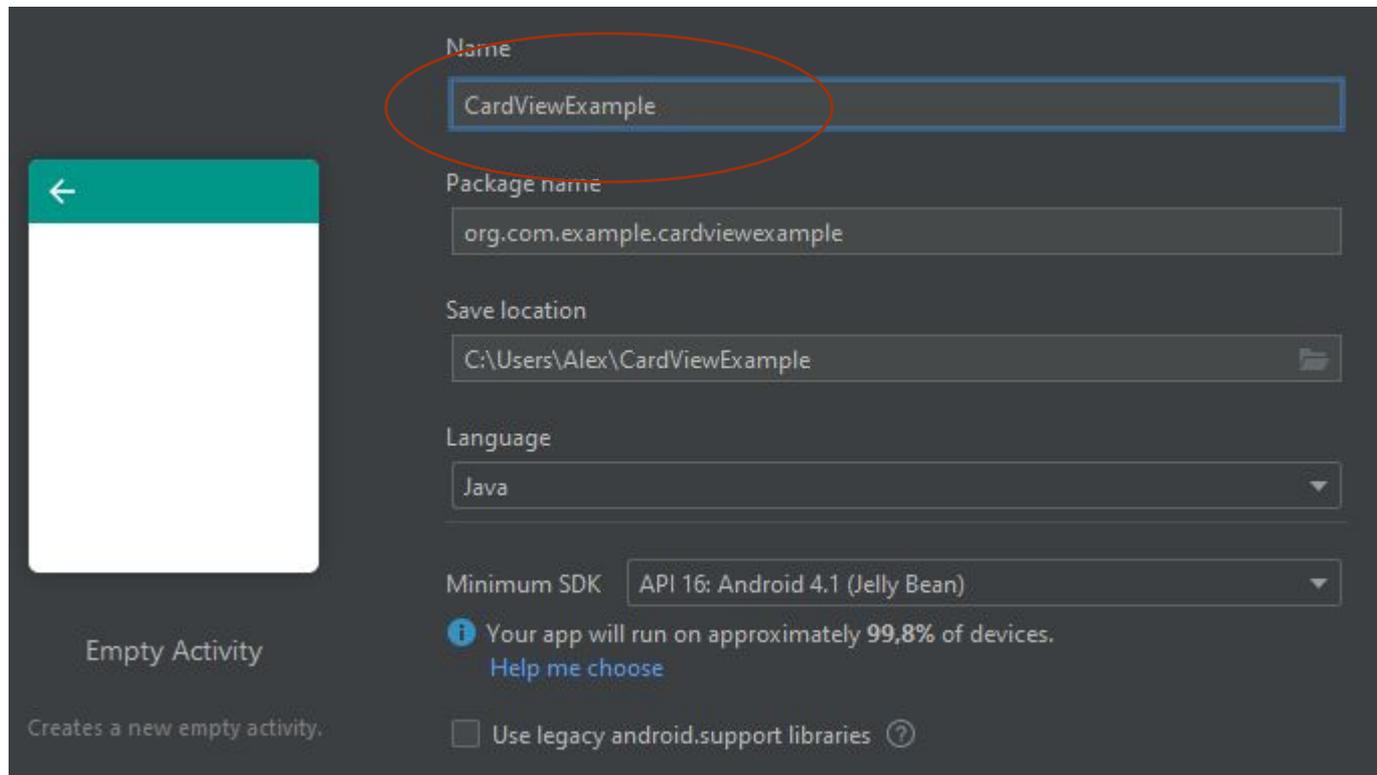


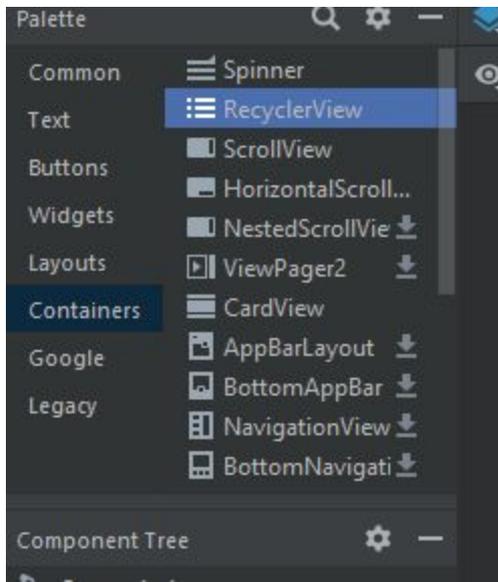
StaggeredGrid



Задача:
Реализовать пример
с RecyclerView и CardView

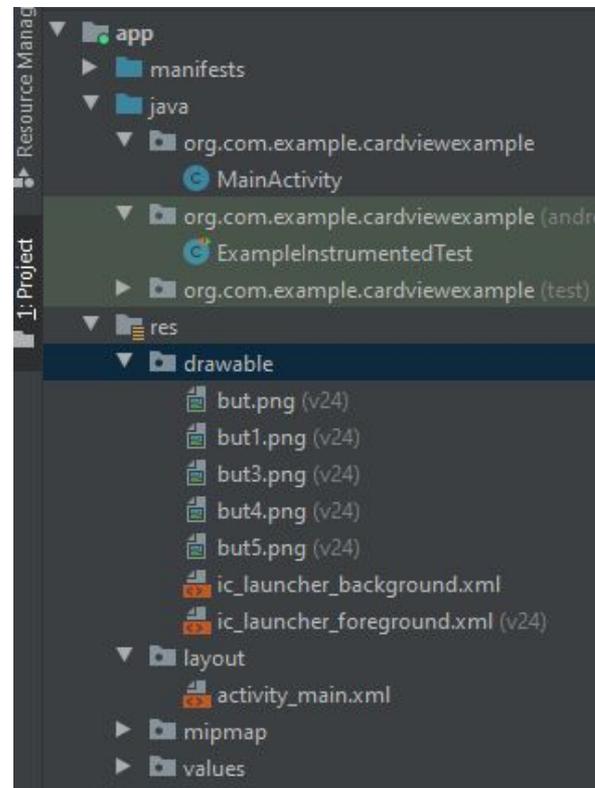
❑ Создать проект



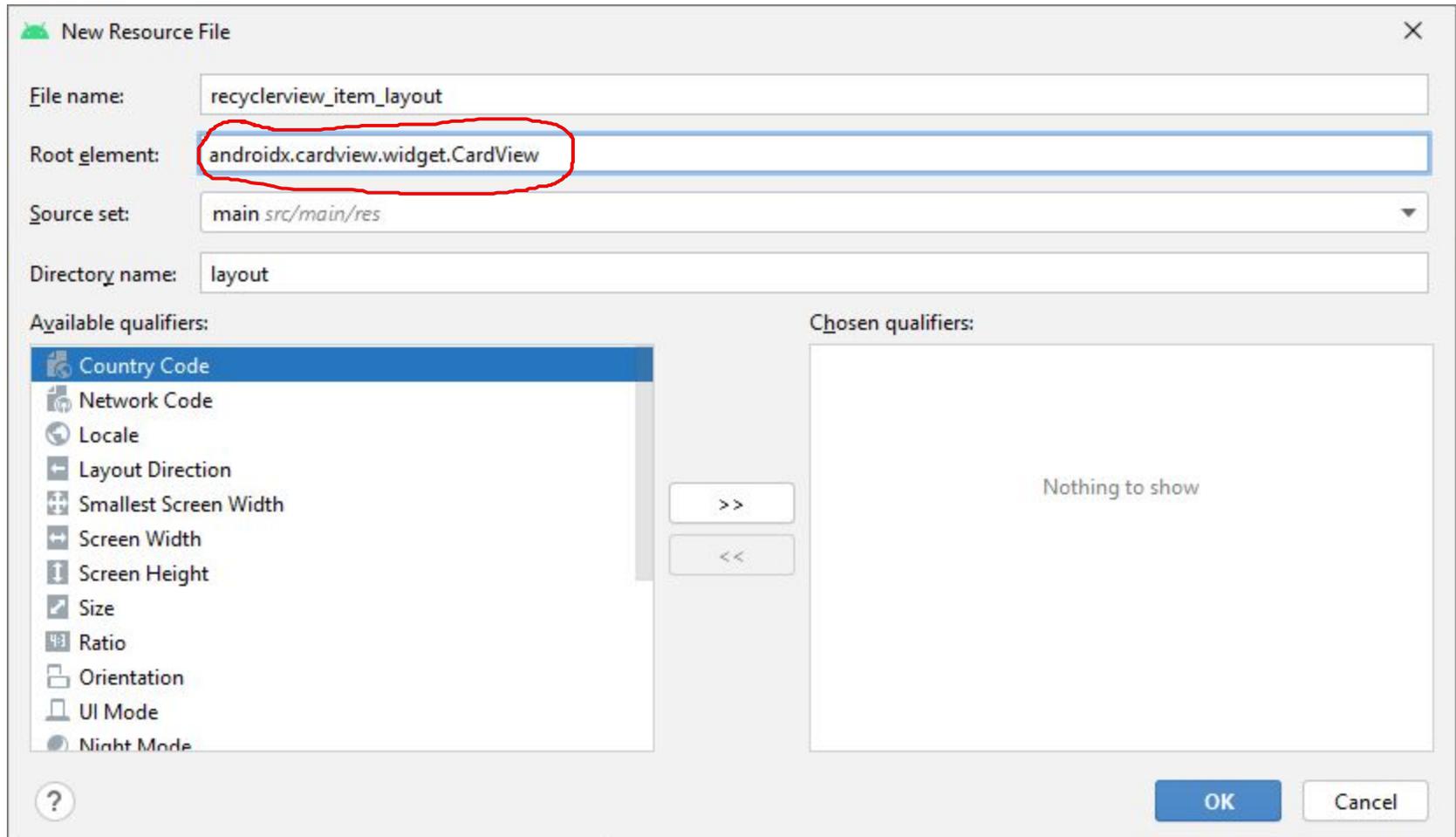


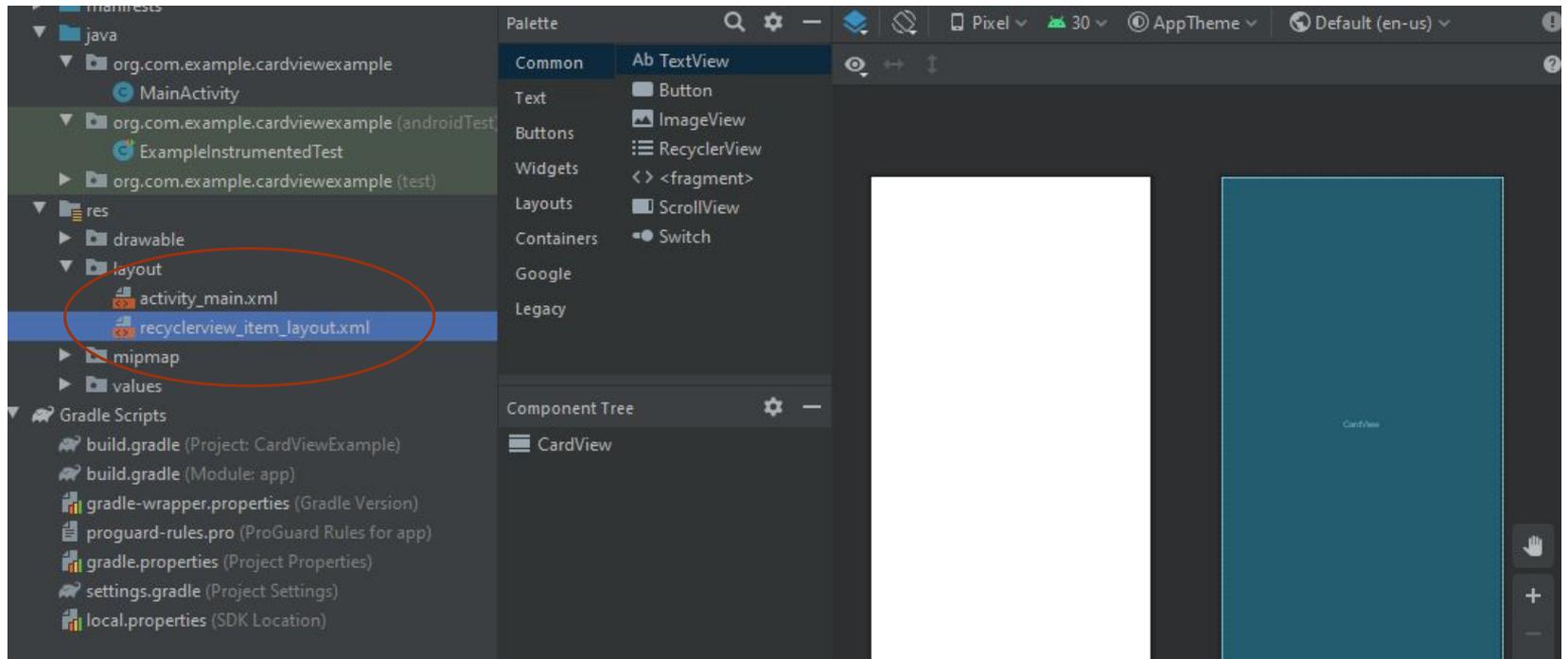
❑ Разместить виджет RecyclerView

❑ Подготовить файлы изображений, скопировать их в папку **drawable** в project:



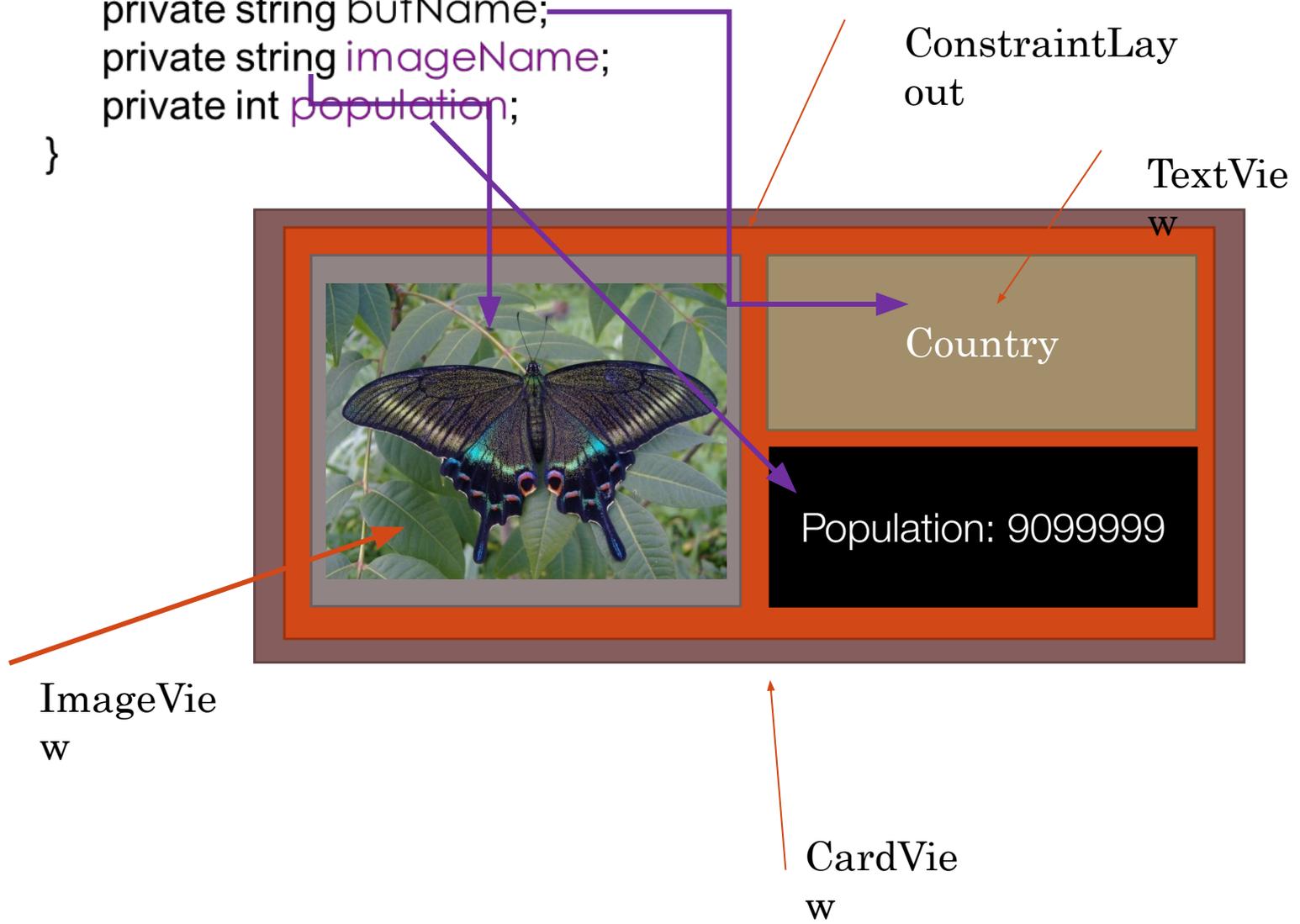
❑ СОЗДАТЬ Layout y RecyclerView Item





Дизайн интерфейса RecyclerView Item:

```
public class Butterfly {  
    private string butName;  
    private string imageName;  
    private int population;  
}
```



- ❑ Необходимо открыть файл `recyclerview_item_layout.xml` и настроить некоторые важные атрибуты для `CardView`:

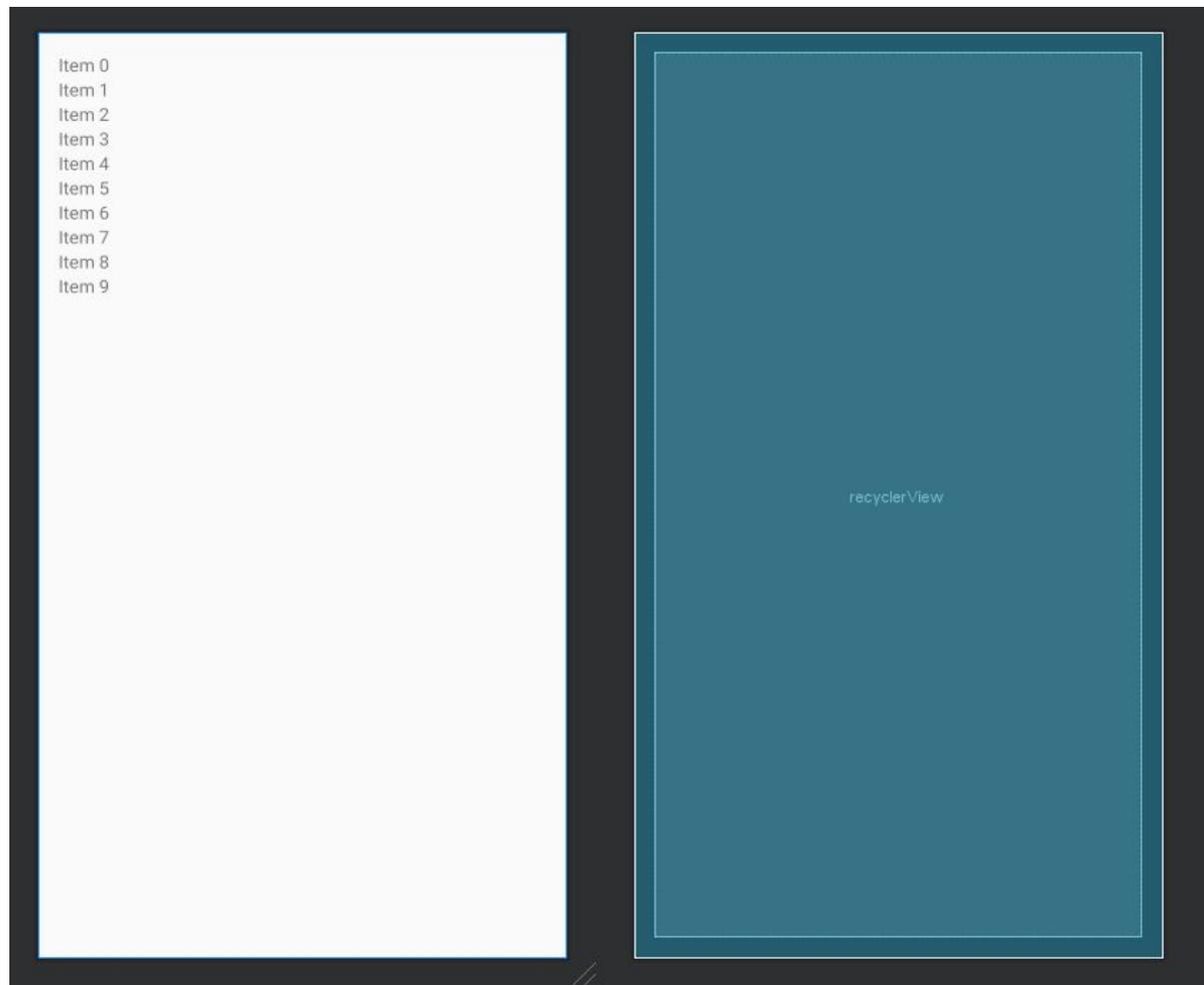
```
<androidx.cardview.widget.CardView
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_margin="5dp"
    app:cardBackgroundColor="@color/white"
    >
```

Доработать дизайн



```
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <ImageView
        android:id="@+id/imageView"
        android:layout_width="110sp"
        android:layout_height="90sp"
        android:layout_marginStart="16dp"
        android:layout_marginLeft="16dp"
        android:layout_marginTop="16dp"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:srcCompat="@drawable/ic_launcher_foreground" />
    <TextView
        android:id="@+id/textView_butName"
        android:layout_width="0dp"
        android:layout_height="35dp"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="16dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:text="Butterfly Name"
        android:textSize="22sp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toTopOf="parent" />
    <TextView
        android:id="@+id/textView_population"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="8dp"
        android:layout_marginLeft="8dp"
        android:layout_marginTop="8dp"
        android:layout_marginEnd="16dp"
        android:layout_marginRight="16dp"
        android:text="Popupation"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toEndOf="@+id/imageView"
        app:layout_constraintTop_toBottomOf="@+id/textView_butName" />
```

- Смоделировать дизайн интерфейса для `activity_main.xml`:

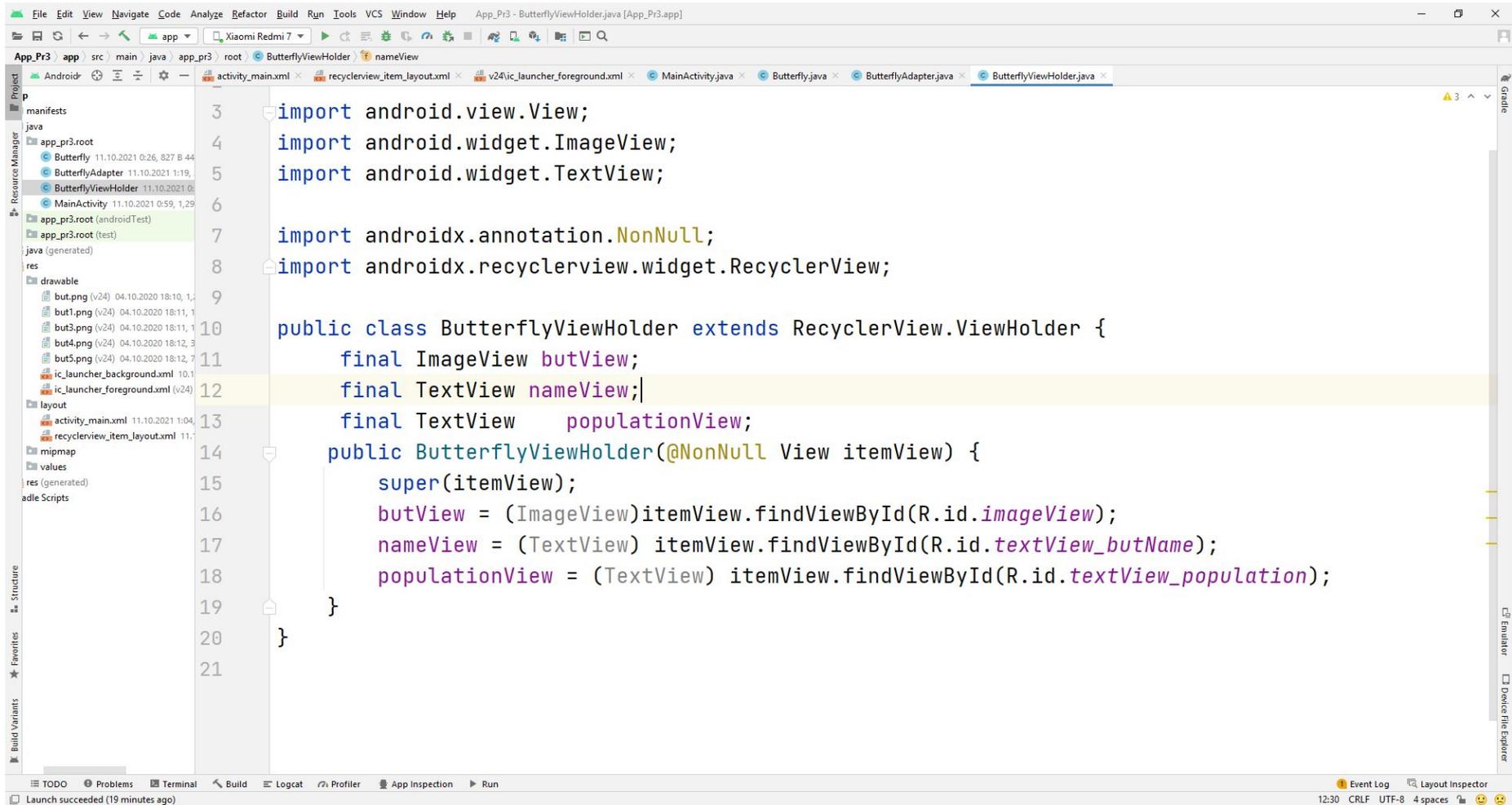


```
<androidx.recyclerview.widget.RecyclerView  
    android:id="@+id/recyclerView"  
  
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:layout_editor_absoluteX="1dp"  
    tools:layout_editor_absoluteY="1dp"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:layout_constraintBottom_toBottomOf="parent"  
    android:layout_margin="10dp"/>
```

Butterfly.java

```
1 package app_pr3.root;
2
3 public class Butterfly{
4     private String butName;
5     private String imageName;
6     private int population;
7
8     public Butterfly(String butName, String imageName, int population) {
9         this.butName = butName;
10        this.imageName = imageName;
11        this.population = population;
12    }
13
14    public String getButName() {
15        return butName;
16    }
17
18    public void setButName(String butName) {
19        this.butName = butName;
20    }
21 }
22
```

ViewHolder



```
3 import android.view.View;
4 import android.widget.ImageView;
5 import android.widget.TextView;
6
7 import androidx.annotation.NonNull;
8 import androidx.recyclerview.widget.RecyclerView;
9
10 public class ButterflyViewHolder extends RecyclerView.ViewHolder {
11     final ImageView imageView;
12     final TextView textView_butName;
13     final TextView textView_population;
14     public ButterflyViewHolder(@NonNull View itemView) {
15         super(itemView);
16         imageView = (ImageView) itemView.findViewById(R.id.imageView);
17         textView_butName = (TextView) itemView.findViewById(R.id.textView_butName);
18         textView_population = (TextView) itemView.findViewById(R.id.textView_population);
19     }
20 }
21
```

The screenshot shows an IDE window titled "App_Pr3 - ButterflyViewHolder.java [App_Pr3.app]". The left sidebar displays the Project, Resource Manager, and Structure views. The main editor area shows the code for ButterflyViewHolder.java. The code includes imports for View, ImageView, TextView, NonNull, and RecyclerView. The class ButterflyViewHolder extends RecyclerView.ViewHolder and has three final fields: imageView, textView_butName, and textView_population. The constructor takes a View itemView and initializes these fields using findViewById. The IDE interface includes a menu bar at the top, a toolbar, and a status bar at the bottom showing "Launch succeeded (19 minutes ago)", "Event Log", "Layout Inspector", and system information like "12:30 CRLF UTF-8 4 spaces".

CustomRecyclerViewAdapter.java

```
1 package app_pr3.root;
2
3 import ...
4
5 public class ButterflyAdapter extends RecyclerView.Adapter<ButterflyViewHolder>{
6     private final LayoutInflater inflater;
7     private final List<Butterfly> butterflyList;
8
9     ButterflyAdapter(Context context, List<Butterfly> butterflyList) {
10
11         this.butterflyList = butterflyList;
12         this.inflater = LayoutInflater.from(context);
13     }
14
15     @NonNull
16     @Override
17     public ButterflyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
18         View view = inflater.inflate(R.layout.recyclerview_item_layout, parent, attachToRoot: false);
19         view.setOnClickListener(v->handleRecyclerViewItemClick((RecyclerView)parent,v));
20         return new ButterflyViewHolder(view);
21     }
22
23     private void handleRecyclerViewItemClick(RecyclerView parent, View v) {
24         int itemPosition=parent.getChildLayoutPosition(v);
25         Butterfly butterfly= butterflyList.get(itemPosition);
26         Toast.makeText(this.inflater.getContext(),butterfly.getButName(),Toast.LENGTH_LONG).show();
27     }
28
29
30
31
32
33
34
35
36
37
```

Launch succeeded (21 minutes ago)

40:5 CRLF UTF-8 4 spaces

CustomRecyclerViewAdapter.java

```
39
40
41 @Override
42 public void onBindViewHolder(@NonNull ButterflyViewHolder holder, int position) {
43     Butterfly butterfly = butterflyList.get(position);
44     int resId=this.getDrowableResIdByName(butterfly.getImageName());
45     holder.butView.setImageResource (resId);
46     holder.nameView.setText(butterfly.getButName());
47     holder.populationView.setText("Популяция: "+butterfly.getPopulation());
48 }
49
50 private int getDrowableResIdByName(String imageName) {
51     String pcgName=inflater.getContext().getPackageName();
52     int resId=inflater.getContext().getResources().getIdentifier(imageName, defType: "drawable", pcgName);
53     return resId;
54 }
55
56 @Override
57 public int getItemCount() {
58     return butterflyList.size();
59 }
60 }
61
```

The screenshot shows an IDE window titled "App_Pr3 - ButterflyAdapter.java [App_Pr3.app]". The code is for a custom RecyclerView adapter. The `onBindViewHolder` method binds data from a `Butterfly` object to a `ButterflyViewHolder`. It sets the image resource, the button name, and the population count. The `getItemCount` method returns the size of the `butterflyList`. The `getDrawableResIdByName` method finds the resource ID for a given image name. The IDE interface includes a Project Explorer on the left, a toolbar at the top, and a status bar at the bottom.

MainActivity.java

```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help App_Pr3 - MainActivity.java [App_Pr3.app]
Xiaomi Redmi 7
App_Pr3 | app | src | main | java | app_pr3 | root | MainActivity | onCreate
activity_main.xml | recyclerView_item_layout.xml | v24/ic_launcher_foreground.xml | MainActivity.java | Butterfly.java | ButterflyAdapter.java | ButterflyViewHolder.java
Project
  app
    manifests
    java
      app_pr3.root
        Butterfly 11.10.2021 0:26
        ButterflyAdapter 11.10.2
        ButterflyViewHolder 11.
        MainActivity 11.10.2021
      app_pr3.root (androidTest)
      app_pr3.root (test)
    java (generated)
    res
      drawable
        but.png (v24) 04.10.202
        but1.png (v24) 04.10.20
        but3.png (v24) 04.10.20
        but4.png (v24) 04.10.20
        but5.png (v24) 04.10.20
        ic_launcher_background
        ic_launcher_foreground
      layout
        activity_main.xml 11.10.
        recyclerView_item_layo
      mipmap
      values
      res (generated)
    Gradle Scripts
Resource Manager
Structure
Favorites
Build Variants
TODO Problems Terminal Logcat Profiler App Inspection
* daemon started successfully (moments ago)

package app_pr3.root;

import ...

public class MainActivity extends AppCompatActivity {
    private RecyclerView recyclerView;
    ArrayList<Butterfly> butterflyList = new ArrayList<>();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initializeData();
        recyclerView=findViewById(R.id.recyclerView);
        // создает адаптер
        ButterflyAdapter adapter = new ButterflyAdapter(this, butterflyList);
        // устанавливаем для списка адаптер
        recyclerView.setAdapter(adapter);
    }

    private void initializeData(){

        butterflyList .add(new Butterfly ("Голубянка", "but",200000));
        butterflyList .add(new Butterfly ("Пестрокрыльница", "but1",345000));
        butterflyList .add(new Butterfly ("Павлиний глаз", "but3", 345620));
        butterflyList .add(new Butterfly ("Монарх", "but4", 2341567));
        butterflyList .add(new Butterfly ("Адмирал", "but5", 8765400));
    }
}
```

