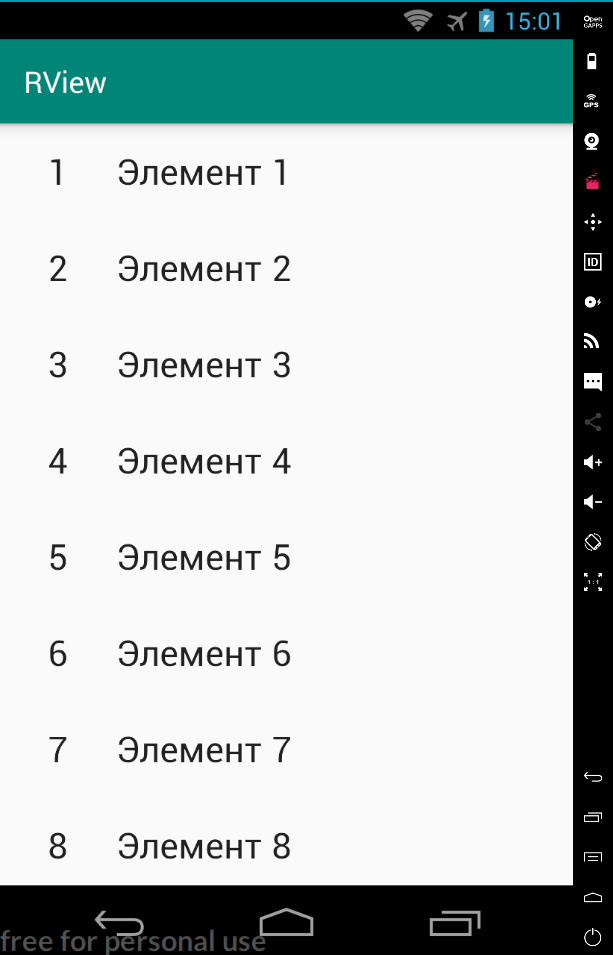
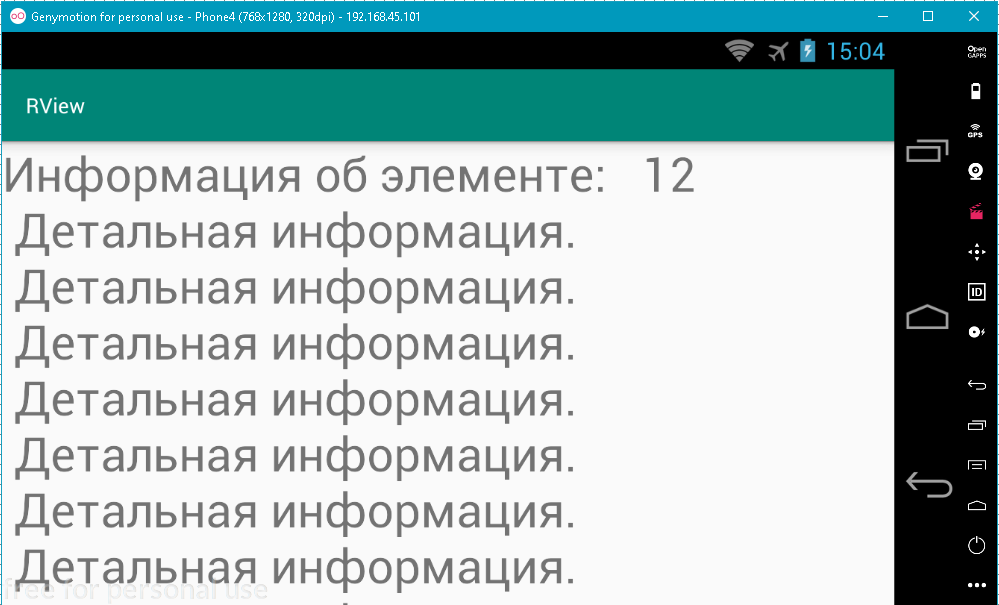
ЛР №3. Создать приложение со списком [**RecyclerView**](https://developer.android.com/guide/topics/ui/layout/recyclerview) (рис.1, 2):

1. Создать приложение с шаблоном **EmptyActivity.**
2. Добавьте класс **DummyContent** и в нём создайте генератор списка с детальной информацией по каждому элементу списка (см.код в конце лабораторной работы). Каждый элемент списка содержит:  
   номер **id**, наименование **content** и детальную информацию **details**.
3. Измените макет стартовой **activity\_main**, удалив текстовое поле и добавив вместо него RecyclerView c id = "@+id/list".
4. Добавьте в файл dimens.xml (если его нет, то создайте его в папке values) размер для текста textSize = 24sp. (При создании текстовых полей во всех макетах задайте атрибут textSize равный добавленному параметру. **Не рекомендуется** задавать числовые значения непосредственно в атрибуты.)
5. Создайте макет для элемента списка **list\_item.xml**, состоящего из 2 текстовых полей (номер **id**, наименование **content**) в соответствии с рис. 1
6. Создайте новую активность **DetailActivity** с макетом **detail\_activity.xml** для подробной информации о выбранном элементе, в соответствии с рис.2. Макет должен содержать одно текстовое поле **details**, содержащее текст на рис.2: «Информация об элементе: и номер элемента k, плюс набор из k строк «Детальная информация».

Рис.1 Рис.2



1. Создайте новый класс для адаптера RecyclerView и создайте в нём:
   * конструктор для адаптера;
   * класс ViewHolder;
   * методы:
     + onCreateViewHolder();
     + onBindViewHolder();
     + getItemCount();
   * метод для обработки выбора элемента из списка RecyclerView (onClick).
2. В случае затруднений смотрите код в конце лабораторной работы.

import java.util.ArrayList;

import java.util.List;

/\*\* Вспомогательный класс для заполнения списка \*/

public **class** **DummyContent** {

/\*\* Списочный массив элементов \*/

public static final List<DummyItem> ITEMS = new ArrayList<DummyItem>();

private static final int COUNT = 25;

static {

// Добавление элементов в список.

for (int k = 1; k <= COUNT; k++) {

StringBuilder builder = new StringBuilder();

builder.append("Информация об элементе: ").append(k);

for (int j = 0; j < k; j++) { builder.append("\n Детальная информация. "); }

ITEMS.add(new DummyItem(String.valueOf(k), "Элемент " + k, builder.toString()));

}

}

/\*\* Создание класса элементов для списка. \*/

public static **class DummyItem** {

public final String id;

public final String content;

public final String details;

public DummyItem(String id, String content, String details) {

this.id = id;

this.content = content;

this.details = details;

}

@Override

public String toString() { return content; }

}

}

public class RViewAdapter extends RecyclerView.Adapter<RViewAdapter.ViewHolder> {  
 private final List<DummyContent.DummyItem> items**;** RViewAdapter(List<DummyContent.DummyItem> items) {  
 this.items = items**;** }  
 @Override  
 public ViewHolder onCreateViewHolder(ViewGroup parent**,** int viewType) {  
 View view = LayoutInflater.*from*(parent.getContext())  
 .inflate(R.layout.*item***,** parent**,** false)**;** return new ViewHolder(view)**;** }  
 @Override  
 public void onBindViewHolder(final ViewHolder holder**,** int position) {  
 holder.idView.setText(items.get(position).id)**;** holder.contentView.setText(items.get(position).content)**;** holder.itemView.setTag(items.get(position))**;** holder.itemView.setOnClickListener(onClickListener)**;** }  
 @Override  
 public int getItemCount() { return items.size()**;** }  
  
 class ViewHolder extends RecyclerView.ViewHolder {  
 final TextView idView**;** final TextView contentView**;** ViewHolder(View view) {  
 super(view)**;** idView = view.findViewById(R.id.*id\_text*)**;** contentView = view.findViewById(R.id.*content*)**;** }  
 }  
  
 final private View.OnClickListener onClickListener = new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 DummyContent.DummyItem item = (DummyContent.DummyItem) view.getTag()**;** Context context = view.getContext()**;** Intent intent = new Intent(context**,** DetailActivity.class)**;** intent.putExtra(*ARG\_ITEM\_ID***,** item.id)**;** context.startActivity(intent)**;** }  
 }**;**}