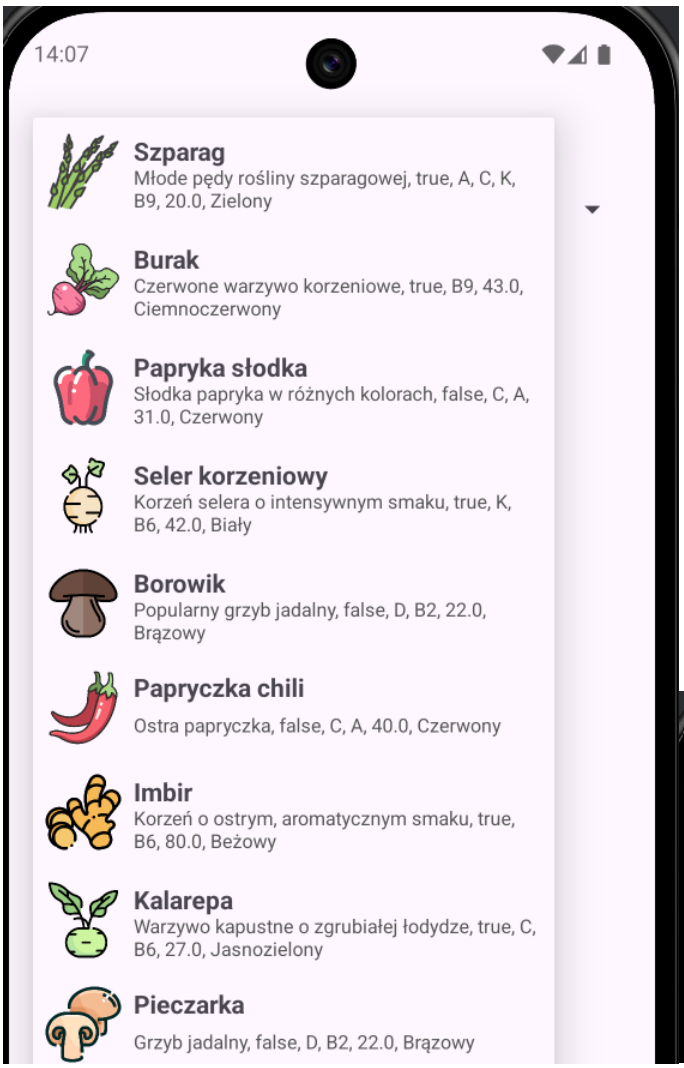
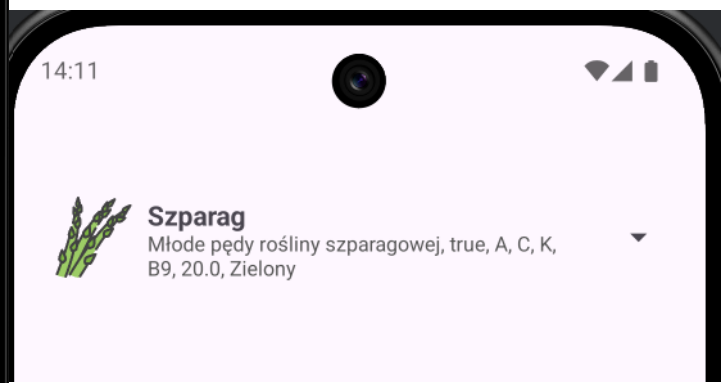
Ćwiczenia 15 – Android studio – BaseAdapter, Spinner

Na koniec zajęć prześlij pliki źródłowe (.xml, .java)+ obrazek do zasobu w teams.

1. Utwórz projekt o nazwie ArrayAdapter na podstawie Empty Activity, dobierz odpowiednie API (min. 26).
2. Otworzyć dokumentację:
   https://developer.android.com/guide/topics/ui/controls/spinner
   https://developer.android.com/guide/topics/resources/string-resource#java
3. Docelowo chcemy uzyskać :

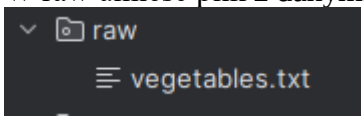




4. Dodaj w activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_marginTop="75dp"
    tools:context=".MainActivity">

    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="16dp"
        tools:ignore="MissingConstraints" />


</LinearLayout>
```

5. W raw umieść plik z danymi:



6. Uzupełnij MainActivity.java:

```
private String[] vegetableNames;
2 usages
private String[] vegetableDescriptions;
3 usages
private int[] vegetableImages;
1 usage
private static final String TAG = "MainActivity";
2 usages
private Spinner spinner;
```

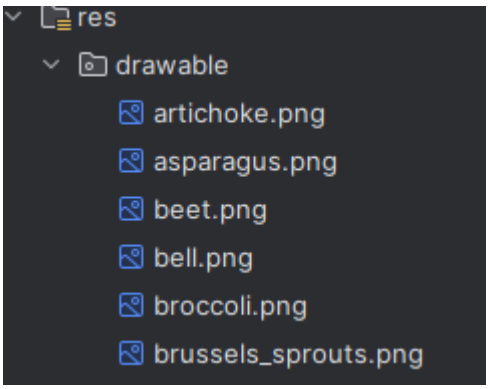
7. Sprawdź czy poprawnie wczytujesz dane, np.:

```
private void loadVegetablesFromRawLog() throws IOException {
    try (InputStream is = getResources().openRawResource(R.raw.vegetables);
         BufferedReader reader = new BufferedReader(new InputStreamReader(is))) {

        String line;
        while ((line = reader.readLine()) != null) {
            // Rozbijamy dane po cudzysłowie
            String[] parts = line.split( regex: "\",\\s*\"");
            if (parts.length >= 3) {
                String name = parts[0].replace( target: "\"", replacement: "");
                String fileName = parts[1].replace( target: "\"", replacement: "");
                String desc = parts[2].replace( target: "\"", replacement: "");
                Log.d(TAG, msg: name + " " + fileName + " " + desc + "\n");
            }
        }

    }
}
```

8. Przetestuj aplikację, uruchom na urządzeniu.

```
D Papryczka chili chili-pepper.png Ostra papryczka, false, C, A, 40.0, Czerwony
D Imbir ginger.png Korzeń o ostrym, aromatycznym smaku, true, B6, 80.0, Beżowy
D Kalarepa kohlrabi.png Warzywo kapustne o zgrubiałej łodydze, true, C, B6, 27.0, Jasnozielony
D Pieczarka mushroom.png Grzyb jadalny, false, D, B2, 22.0, Brązowy
```

9. Dodaj pliki png do drawable



10. Utwórz tablicę dla plików:

```
private void loadVegetablesFromRaw() throws IOException {
    List<String> names = new ArrayList<>();
    List<String> descriptions = new ArrayList<>();
    List<Integer> images = new ArrayList<>();

    try (InputStream is = getResources().openRawResource(R.raw.vegetables);
         BufferedReader reader = new BufferedReader(new InputStreamReader(is))) {
        String line;

        while ((line = reader.readLine()) != null) {
            String[] parts = line.split( regex: "\",\\s*\"");
            if (parts.length >= 3) {
                String name = parts[0].replace( target: "\"", replacement: "").trim();
                String imageFile = parts[1].replace( target: "\"", replacement: "").trim().replace( target: ".png", replacement:
                String description = parts[2].replace( target: "\"", replacement: "").trim();

                int imageResId = getResources().getIdentifier(imageFile, defType: "drawable", getPackageName());
                if (imageResId == 0) {
                    imageResId = R.drawable.bug;
                }
                names.add(name);
                descriptions.add(description);
                images.add(imageResId);
            }
        }

        vegetableNames = names.toArray(new String[0]);
        vegetableDescriptions = descriptions.toArray(new String[0]);
        vegetableImages = new int[images.size()];
        for (int i = 0; i < images.size(); i++) {
            vegetableImages[i] = images.get(i);
        }
    }
```

11. Dalej, szkielet dla onCreate():

```
spinner = findViewById(R.id.spinner);

try {
    loadVegetablesFromRaw();
} catch (IOException e) {
    throw new RuntimeException(e);
}

MyAdapter adapter = new MyAdapter( context: this, vegetableNames, vegetableDescriptions, vegetableImages);
spinner.setAdapter(adapter);
```

12. Utwórz plik spinner_item.xml:

```
<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:padding="8dp">

    <ImageView
        android:id="@+id/imageView"
        android:layout_width="48dp"
        android:layout_height="48dp"
        android:scaleType="centerCrop"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        android:contentDescription="@string/app_name" />

    <TextView
        android:id="@+id/nameView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="16sp"
        app:layout_constraintStart_toEndOf="@id/imageView"
        app:layout_constraintTop_toTopOf="@id/imageView"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginStart="8dp" />

    <TextView
        android:id="@+id/descView"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:textSize="12sp"
        android:textColor="#555"
        app:layout_constraintStart_toStartOf="@id/nameView"
        app:layout_constraintTop_toBottomOf="@id/nameView"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="@id/imageView" />
```

13. Utwórz klasę MyAdapter.java, rozrzesz ją o BaseAdapter:

```
public class MyAdapter extends BaseAdapter {

    1 usage
    private Context context;
    4 usages
    private String[] names;
    2 usages
    private String[] descriptions;
    2 usages
    private int[] images;
    2 usages
    private LayoutInflater inflater;

    1 usage
    public MyAdapter(Context context, String[] names, String[] descriptions, int[] images) {
        this.context = context;
        this.names = names;
        this.descriptions = descriptions;
        this.images = images;
        this.inflater = LayoutInflater.from(context);
    }
```

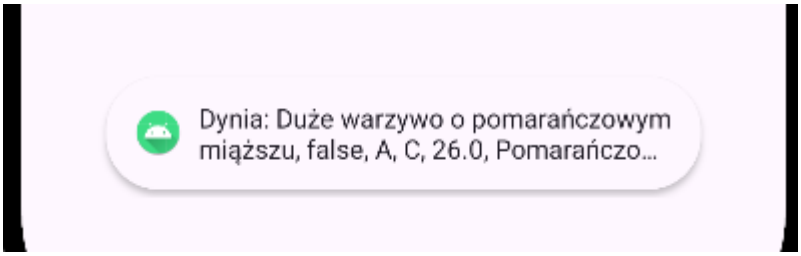
14. Sprawdź wbudowane metody:

```
@Override
public int getCount() {
    return names.length;
}

@Override
public Object getItem(int position) {
    return names[position];
}

@Override
public long getItemId(int position) {
    return position;
}

3 usages
static class ViewHolder {
    2 usages
    ImageView imageView;
    2 usages
    TextView nameView;
    2 usages
    TextView descView;
}
```

15. Ostatnia metoda:

```java
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    ViewHolder holder;

    if (convertView == null) {
        convertView = inflater.inflate(R.layout.spinner_item, parent, attachToRoot: false);
        holder = new ViewHolder();
        holder.imageView = convertView.findViewById(R.id.imageView);
        holder.nameView = convertView.findViewById(R.id.nameView);
        holder.descView = convertView.findViewById(R.id.descView);
        convertView.setTag(holder);
    } else {
        holder = (ViewHolder) convertView.getTag();
    }

    holder.imageView.setImageResource(images[position]);
    holder.nameView.setText(names[position]);
    holder.descView.setText(descriptions[position]);

    return convertView;
}
```

16. Wykonaj zadania:
    a) dodaj obsługę kliknięcia w item, wyświetl toast



```
spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
    2 usages
    boolean isFirstSelection = true;

    no usages
    @Override
    public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
        // Opcjonalnie pomijamy pierwszy automatyczny wybór
        if (isFirstSelection) {
            isFirstSelection = false;
            return;
        }

        String name = vegetableNames[position];
        String desc = vegetableDescriptions[position];
        Toast.makeText(parent.getContext(), text: name + ": " + desc, Toast.LENGTH_SHORT).
    }
```

b) utwórz przycisk dodający nowy element do listy

c) zachowaj całą listę w ShredPreferences, sprawdź stan linty po rotacji urządzenia

d) przebuduj projekt tworząc klasę i listę, utwórz stosowny adapter

```
private List<VegetableItem> vegetableItemList;
```

```
public class VegetableItem {
    2 usages
    private String name;
    2 usages
    private String imageFileName;
    2 usages
    private String description;
```

```
public class MyAdapter extends BaseAdapter {

    3 usages
    private final Context context;
    4 usages
    private final List<VegetableItem> items;
    2 usages
    private final LayoutInflater inflater;
```

Obsługa dla listenera:

```
VegetableItem selectedItem = vegetableItemList.get(position);
Toast.makeText(
        context: MainActivity.this,
        text: selectedItem.getName() + ": " + selectedItem.getDescription(
        Toast.LENGTH_SHORT
).show();
```

Metoda czytająca dane:

```
private List<VegetableItem> loadVegetablesItemFromRaw() {
    List<VegetableItem> list = new ArrayList<>();
```

17. KONIEC.