

Компоненты CalendarView, SeekBar

CalendarView

Компонент CalendarView находится в разделе Widgets и выводит на экран календарь.

```
<CalendarView
    android:id="@+id/calendarView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/textView" />
```

Выбранную дату можно отслеживать через метод `setOnDateChangeListener()`:

Пример. Выводит выбранную дату с помощью Toast

```
CalendarView calendarView = (CalendarView) findViewById(R.id.calendarView);
calendarView.setOnDateChangeListener(new CalendarView.OnDateChangeListener() {
    @Override
    public void onSelectedDayChange(CalendarView view, int year,
                                    int month, int dayOfMonth) {
        String selectedDate = dayOfMonth + "." + (month + 1) + "." + year;
        Toast.makeText(getApplicationContext(), selectedDate, Toast.LENGTH_LONG).show();
    }
});
```

SeekBar

SeekBar - это слайдер, когда пользователь может передвигать ползунок пальцем на экране или двигать при помощи клавиш-стрелок.

```
<SeekBar
    android:id="@+id/seekBar"
    android:layout_width="310dp"
    android:layout_height="21dp"
    android:max="100"
    android:progress="10"
```

Атрибут `android:progress` позволяет задать текущее значение ползунка, а атрибут `android:max` - максимально возможное значение.

Метод `setOnSeekBarChangeListener()` позволяет установить обработчики событий изменения значения ползунка.

Пример. Значение ползунка SeekBar отображается в TextView

```

SeekBar seekBar = findViewById(R.id.seekBar);
seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int i, boolean fromUser)
    {
        textView.setText(String.valueOf(i));
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {
    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {
    }
});

```

В метод `setOnSeekBarChangeListener()` передается объект `SeekBar.OnSeekBarChangeListener`, который позволяет установить три метода-обработчика:

- `onProgressChanged`: срабатывает при перетаскивании ползунка по шкале. Передаваемый в метод параметр `progress` позволяет получить новое значение ползунка, которое в данном случае передается в `TextView` для отображения на экране
- `onStartTrackingTouch`: срабатывает при начале перетаскивания ползунка по шкале
- `onStopTrackingTouch`: срабатывает при завершении перетаскивания ползунка по шкале

Задание 1. Идеальный вес.

- `CalendarView` пользователь вводит дату рождения.
- `SeekBar` рост и вес
- `ImageView` – пол человека

Посчитать идеальный вес

The screenshot shows a mobile application interface for calculating ideal weight. The screen is titled "Your data:" and contains several sections:

- Gender:** Two circular icons representing a man and a woman.
- Birthday:** A calendar view showing the month of January 1970. The date "2" is selected and highlighted in a green circle.
- Height:** A horizontal slider control with a green dot indicating the current value.
- Weight:** A horizontal slider control with a green dot indicating the current value.
- CALCULATE:** A large button at the bottom of the screen.

Списки

При работе со списками мы имеем дело с тремя компонентами.

- Во-первых, это сами элементы списков (ListView, GridView, RecyclerView), которые отображают данные.
- Во-вторых, это источник данных - массив, объект ArrayList, база данных и т.д., в котором находятся сами отображаемые данные.
- И в-третьих, это адаптеры - специальные компоненты, которые связывают источник данных с элементом списка.

ArrayAdapter

Класс ArrayAdapter представляет собой простейший адаптер, который связывает массив данных с набором элементов TextView, из которых, к примеру, может состоять ListView.

Пример. Список стран

```
<ListView
    android:id="@+id/listViewCountries"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
</ListView>
```

```
public class MainActivity extends AppCompatActivity {

    // набор данных, которые свяжем со списком
    String[] countries = { "Бразилия", "Аргентина", "Колумбия", "Чили", "Уругвай" };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // получаем элемент ListView
        ListView countriesList = (ListView) findViewById(R.id.countriesList);

        // создаем адаптер
        ArrayAdapter<String> adapter = new ArrayAdapter(this,
            android.R.layout.simple_list_item_1, countries);

        // устанавливаем для списка адаптер
        countriesList.setAdapter(adapter);
    }
}
```

- `this` : текущий объект activity
- `android.R.layout.simple_list_item_1` : файл разметки списка, который фреймворк представляет по умолчанию. Можно создать свою разметку и потом в коде изменить id на id нужной разметки
- `countries`: массив данных.

Подобный массив строк гораздо удобнее было бы хранить в файле xml в виде ресурса (res/values/filename, например countries).

Ресурсы массивов строк представляют элемент типа string-array. Они могут находиться в каталоге res/values в xml-файле с произвольным именем.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="имя_массива_строк">
        <item>элемент</item>
    </string-array>
</resources>
```

Все элементы массива представляют набор значений <item>

```
// получаем ресурс
String[] countries = getResources().getStringArray(R.array.countries);
```

Но необязательно добавлять список строк в ListView программно. У элемента ListView есть атрибут entries, который в качестве значения может принимать ресурс string-array

```
<ListView
    android:entries="@array/countries"
```

Для выбора элемента в ListView устанавливается слушатель OnItemClickListener. Этот слушатель имеет один метод onItemClick, через параметры которого можно получить выделенный элемент и его данные.

```
// получаем элемент TextView
TextView textViewSelection = (TextView) findViewById(R.id.textViewCountry);
// получаем элемент ListView
ListView countriesList = (ListView) findViewById(R.id.countriesList);

// создаем адаптер
ArrayAdapter<String> adapter = new ArrayAdapter(this,
    android.R.layout.simple_list_item_1, countries);
// устанавливаем для списка адаптер
countriesList.setAdapter(adapter);
// добавляем для списка слушатель
countriesList.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View v, int position, long id)
    {
        // по позиции получаем выбранный элемент
        String selectedItem = countries[position];
        // установка текста элемента TextView
        textViewSelection.setText(selectedItem);
    }
});
```

Задание 2

Тест Столицы мира. Добавьте в проект фотографии столиц. Фотографии должны отображаться в соответствии с вопросом. Ответы отображаются в списке из 4-х элементов.