

Методические указания к Практической работе №4

по дисциплине «Разработка кроссплатформенных мобильных приложений»

Тема работы: Виджеты ListView, ListTile, SnackBar

План практической работы:

- Знакомство с прокручиваемым списком ListView и виджетом отображения элемента списка ListTile
- Знакомство с виджетом отображения короткого уведомления SnackBar
- Выполнение практической работы №4 в соответствии с заданием

Последовательность выполнения практической работы:

1. Знакомство с прокручиваемым списком ListView и виджетом отображения элемента списка ListTile

Виджет ListView (Список)

ListView используется для отображения прокручиваемого списка элементов (например, список сообщений в чате, список дел или галерея изображений и т.д.).

Можно создавать вертикальные списки, горизонтальные списки или пользовательские макеты для каждого элемента в списке.

Базовое использование с помощью конструктора ListView() – по умолчанию

```
Widget _myListView() {  
  return ListView(  
    padding: const EdgeInsets.all(16.0),  
    children: <Widget> [  
      Text('Algorithm', style: TextStyle(fontSize: 20)),  
      Text('Database', style: TextStyle(fontSize: 20)),  
      Text('Operating System (OS)', style: TextStyle(fontSize: 20)),  
      Text('Artificial Intelligence (AI)', style: TextStyle(fontSize: 20)),  
      Text('Cloud Computing', style: TextStyle(fontSize: 20)),  
      Text('Computer Network', style: TextStyle(fontSize: 20)),  
      Text('Cybersecurity', style: TextStyle(fontSize: 20)),  
      Text('Programming Language', style: TextStyle(fontSize: 20)),  
      Text('Big Data', style: TextStyle(fontSize: 20)),  
      Text('Virtual Reality (VR)', style: TextStyle(fontSize: 20)),  
      Text('Machine Learning (ML)', style: TextStyle(fontSize: 20)),  
      Text('Computer Virus', style: TextStyle(fontSize: 20)),  
      Text('Binary System', style: TextStyle(fontSize: 20)),  
      Text('Data Encryption', style: TextStyle(fontSize: 20)),  
    ], //<Widget>[] <Widget>[]  
  ); //ListView ListView  
}
```

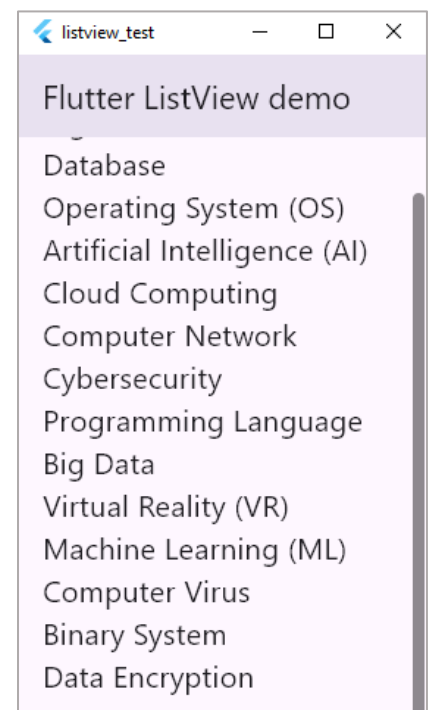


Рисунок 1 – Приложение, выводящее список ИТ-терминов

Применение базового **ListView()** предполагает, что **все** элементы списка будут отрендерены (выведены на экран) сразу же, а не в момент их появления на экране, что снижает производительность приложения.

ListView.builder() отрисовывает только те элементы, которые видны на экране, и несколько элементов до и после видимой области.

Параметры:

- itemCount – количество элементов в списке (указывает Flutter, сколько всего элементов нужно отобразить);
- itemBuilder: (context, index) { return Text(list[index]); } – отрисовывает один элемент списка по его индексу (он вызывается itemCount раз)

Дополнительные параметры:

- `shrinkWrap` — (true/false) определяет растяжение по главной оси
- `scrollDirection: Axis.horizontal` — определяет горизонтальный `ListView`

Конструктор **`ListView.separated`** добавляет разделители между элементами (линии, отступы, иконки, текст)

Дополнительный параметр `separatorBuilder` (разделитель):

Линия – `Divider()`

Текст – `Text()`

Отступ – `SizedBox()/Padding()`

Иконка – `Icon()`

Параметр `scrollDirection`

По умолчанию `ListView` (список) прокручивается по вертикали.

Для того, чтобы список прокручивался горизонтально, необходимо изменить направление прокрутки на горизонтальное: `scrollDirection: Axis.horizontal`

Но нужно учитывать, что в случае горизонтальной прокрутки `ListView` не может определить ширину дочерних элементов. Для решения этой проблемы нужно указать ширину дочерних элементов. Можно просто, к примеру, обернуть все наши элементы в контейнер и задать им ширину или воспользоваться таким свойством `ListView` как `itemExtent`.

Виджет `ListTile`

`ListTile` — это готовый виджет Flutter для отображения одной строки списка с типичной структурой:

- иконка/аватарка слева,
- заголовок,
- подзаголовок,
- иконка/кнопка справа.

```
ListTile(
  leading: Icon(Icons.computer),           // иконка/аватарка слева
  title: Text("Заголовок"),                // основной текст
  subtitle: Text("Подзаголовок"),         // подзаголовок
  trailing: Icon(Icons.arrow_forward),     // иконка справа
  onTap: () {                             // действие по нажатию
    print("Tapped!");
  },
)
```

Пример приложения для вывода ИТ-терминов с использованием `ListView.builder()` и `ListTile()`
Программный код:

```
import 'package:flutter/material.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: Scaffold(
        appBar: AppBar(title: const Text('Informatics terms')),
        body: const TermList(),
      ),
    );
  }
}

class TermList extends StatelessWidget {
```

```

const TermList({super.key});

// Список понятий
final List<String> terms = const [
  "Algorithm",
  "Database",
  "Operating System",
  "Artificial Intelligence",
  "Cloud Computing",
  "Computer Network",
  "Cybersecurity",
  "Programming Language",
  "Big Data",
  "Virtual Reality",
  "Machine Learning",
  "Computer Virus",
  "Software Development Life Cycle",
  "Binary System",
  "Data Encryption"
];

// Короткие описания для subtitle
final List<String> descriptions = const [
  "Step-by-step procedure to solve a problem",
  "Organized collection of data",
  "Software that manages hardware",
  "Simulation of human intelligence by machines",
  "Computing via the internet",
  "Interconnected computers sharing data",
  "Protecting systems from attacks",
  "Language for writing programs",
  "Very large data sets",
  "Computer-generated immersive world",
  "Learning from data (AI subset)",
  "Malicious program",
  "Software creation process",
  "Base-2 number system",
  "Encoding data for security"
];

@override
Widget build(BuildContext context) {
  return ListView.builder(
    itemCount: terms.length,
    itemBuilder: (context, index) {
      return ListTile(
        leading: const Icon(Icons.computer, color: Colors.blue), // слева
        title: Text(terms[index]), // текст
        trailing: const Icon(Icons.arrow_forward_ios, color: Colors.grey), // справа
        onTap: () {
          //действия по клику
        },
      );
    },
  );
}

```

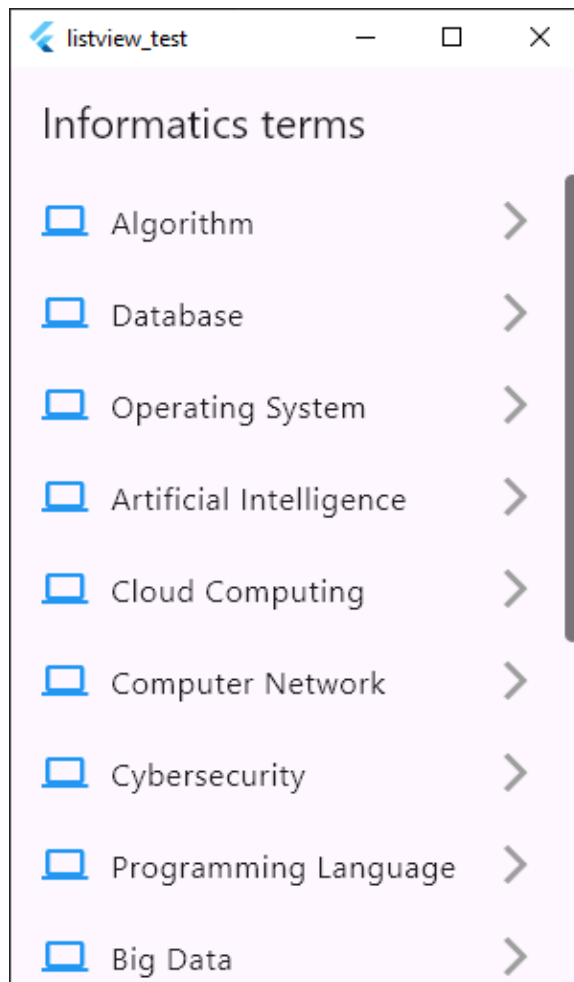


Рисунок 2 – Приложение с ListView и ListTile

2. Знакомство с виджетом отображения короткого уведомления **SnackBar**

Виджет **SnackBar**

SnackBar — это короткое уведомление внизу экрана в стиле Material Design. Показывает краткое сообщение с необязательной кнопкой действия (например, UNDO) и автоматически скрывается через несколько секунд.

Для отображения SnackBar используется ScaffoldMessenger (пример):

```
ScaffoldMessenger.of(context).showSnackBar(
  SnackBar( content: const Text('Item saved'),
    action: SnackBarAction(
      label: 'UNDO',
      onPressed: () {
        // обработка отмены
      },
    ),
    duration: const Duration(seconds: 3),
  ), );
```

3. Задание на практическую работу

Изменить приложение, созданное на предыдущей практической работе (примерный макет ниже), выполнив следующие действия:

- 1) Изображения вывести со скругленными углами в виде горизонтального ListView под описанием ПО
- 2) Список пунктов вывести ниже в виде вертикального ListView в форме карточек (Card) с короткими описаниями и иконками перед и после текста
- 3) При нажатии на карточке выводится уведомление SnackBar с названием выбранного пункта

Примерный макет измененного приложения:

