

Практика 3 (Flutter) – Создание приложения «Список дел»

В ходе практической работы мы реализуем объект «ListView», создадим функционал для удаления элементов и сделаем всплывающее окно для добавления новых заданий в список.

1. Установка Flutter SDK

- Сначала установите Flutter SDK. Перейдите на [официальный сайт Flutter](#) и выберите вашу операционную систему (Windows, macOS, Linux) для получения подробных инструкций по установке.
- После установки Flutter SDK, убедитесь, что у вас установлен Dart SDK. Обычно он устанавливается вместе с Flutter.
- Откройте терминал или командную строку и выполните команду `flutter doctor` чтобы проверить, все ли установлено корректно.

2. Создание нового Flutter проекта

В терминале выполните команду:

```
flutter create имя_вашего_проекта
```

Перейдите в каталог вашего проекта:

```
cd имя_вашего_проекта
```

3. Добавление зависимостей

Откройте файл `pubspec.yaml`, который находится в корне вашего проекта.

Добавьте необходимые зависимости в раздел `dependencies`. Для базовых библиотек Flutter это не требуется, но если у вас есть сторонние пакеты, добавьте их.

Запустите `flutter pub get` чтобы установить зависимости.

4. Написание кода

Откройте файл `lib/main.dart` в вашем редакторе кода.

Импортируйте необходимые библиотеки:

```
import "package:flutter/material.dart";  
import 'package:flutter_todo/pages/home.dart'; // убедитесь, что этот файл  
существует
```

Добавьте ваш код:

```
void main() => runApp(MaterialApp(  
  theme: ThemeData(  
    primaryColor: Colors.deepOrangeAccent,  
  ),  
  home: Home(),  
));
```

5. Запуск приложения

Запустите симулятор или подключите реальное устройство.

В терминале, находясь в каталоге вашего проекта, выполните команду:

```
flutter run
```

Теперь ваше приложение должно успешно запуститься на выбранном устройстве.

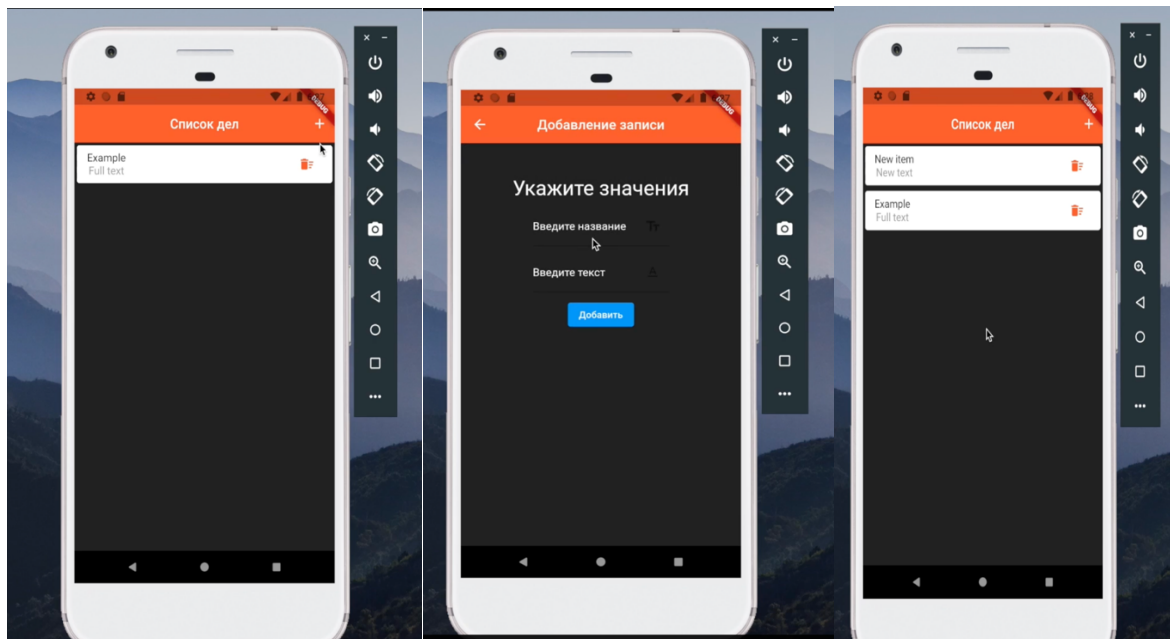
Самостоятельное задание 1. Вывод информации

Выведите на экран текстовую надпись: «Это мое первое приложение».

Приложение Список дел:

Вам необходимо разработать усовершенствованное приложение для работы со списком дел.

Пример реализации нового проекта представлен ниже:



На главной странице реализуйте вывод всех записей. Теперь у каждой записи будет название и дополнительный текст. Подобную реализацию можно сделать используя ряд или столбец при выводе текстовых данных в списке.

Переход на вторую страницу происходит по нажатию иконки в шапке приложения.

На второй странице должна быть форма, что состоит из двух полей. При нажатии на кнопку добавить новая запись должна помещаться в базу данных Firebase, а также окно с формой должно закрываться.

Помощь: при добавлении текстовых полей используйте класс **TextField**. Он не будет корректно добавляться до тех пор, пока вы не поместите его в объект **Container**, в котором будет указана ширина (свойство **width**).

Main.dart:

```
main.dart X
Users > arailymtleubayeva > Downloads > lib > main.dart
1  import 'package:flutter/material.dart';
2  // Подключени всех сцен
3  import 'package:flutter_todo/pages/add_item.dart';
4  import 'package:flutter_todo/pages/main_screen.dart';
5
6  import 'package:firebase_core/firebase_core.dart';
7
8  void main() async {
9    WidgetsFlutterBinding.ensureInitialized();
10   await Firebase.initializeApp();
11
12   runApp(MaterialApp(
13     theme: ThemeData(
14       primaryColor: Colors.deepOrangeAccent,
15     ),
16     initialRoute: '/',
17     routes: {
18       // Маршрутизация
19       '/': (context) => MainScreen(),
20       '/add': (context) => AddItem(),
21     },
22   ));
23 }
24
```

add_item.dart:

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

class AddItem extends StatefulWidget {
  const AddItem({Key key}) : super(key: key);

  @override
  _AddItemState createState() => _AddItemState();
}

class _AddItemState extends State<AddItem> {
  String _title, _text;

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[900],
      // Шапка
      appBar: AppBar(
```

```

        title: Text('Добавление записи'),
        centerTitle: true,
    ),
    body: Container(
        margin: EdgeInsets.only(top: 50),
        child: Row(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
                Column(
                    children: [
                        Text('Укажите значения', style: TextStyle(color: Colors.white,
fontSize: 30)),
                        Padding(padding: EdgeInsets.only(top: 10)),
                        // Для текстовых полей нужно указывать ширину
                        // по этой причине тут создан контейнер и указана в нем ширина
                        Container(
                            width: 200,
                            child: TextField(
                                style: TextStyle(color: Colors.white),
                                decoration: InputDecoration(
                                    border: UnderlineInputBorder(),
                                    labelText: 'Введите название',
                                    labelStyle: new TextStyle(
                                        color: Colors.white,
                                    ),
                                ),
                                suffixIcon: Icon(Icons.text_fields),
                            ),
                            onChanged: (value) {
                                // При вводе информации сохраняем ее в поле класса
                                setState(() {
                                    _title = value;
                                });
                            },
                        ),
                    ],
                ),
                Padding(padding: EdgeInsets.only(top: 10)),
                // Аналогичное второе поле
                Container(
                    width: 200,
                    child: TextField(
                        style: TextStyle(color: Colors.white),
                        decoration: InputDecoration(
                            border: UnderlineInputBorder(),
                            labelText: 'Введите текст',
                            labelStyle: new TextStyle(
                                color: Colors.white,
                            ),
                        ),
                        suffixIcon: Icon(Icons.text_format),
                    ),
                    onChanged: (value) {
                        setState(() {

```

```

        // При вводе информации сохраняем ее в поле класса
        _text = value;
    });
    },
  ),
  Padding(padding: EdgeInsets.only(top: 10)),
  // Кнопка добавления
  ElevatedButton(onPressed: () {
    // При нажатии сохраняем новую запись
    // В новой записе указываем название и указываем текст
    FirebaseFirestore.instance.collection('items').add({'title':
_title, 'text': _text});
    Navigator.pop(context); // Скрываем окно
  }, child: Text('Добавить'))
],
),
],
),
)
);
}
}

```

main_screen.dart:

```

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

class MainScreen extends StatefulWidget {
  const MainScreen({Key key}) : super(key: key);

  @override
  _MainScreenState createState() => _MainScreenState();
}

class _MainScreenState extends State<MainScreen> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.grey[900],
      appBar: AppBar(
        title: Text('Список дел'),
        centerTitle: true,
        actions: [
          // Создаем иконку в шапке
          IconButton(
            icon: Icon(Icons.add),
            onPressed: () {

```

```

        // При нажатии переходим на другую страницу
        Navigator.pushNamed(context, '/add');
    },
)
],
),
body: StreamBuilder( // Вывод всех записей
    stream: FirebaseFirestore.instance.collection('items').snapshots(),
    builder: (BuildContext context, AsyncSnapshot<QuerySnapshot> snapshot) {
        // Здесь проверка на количество элементов
        if(snapshot.data.docs.length == 0) return Center(child: Text('Нет
записей', style: TextStyle(color: Colors.white),));
        return ListView.builder(
            itemCount: snapshot.data.docs.length,
            itemBuilder: (BuildContext context, int index) {
                return Dismissible(
                    key: Key(snapshot.data.docs[index].id),
                    child: Card(
                        child: ListTile(
                            title: Row( // Вывод не просто текста, а целого ряда
                                mainAxisAlignment: MainAxisAlignment.start,
                                children: [
                                    Column(
                                        children: [
                                            // Вывод названия и полного текста
                                            Text(snapshot.data.docs[index].get('title')),
                                            Text(snapshot.data.docs[index].get('text'), style:
TextStyle(color: Colors.grey),),
                                        ],
                                    )
                                ],
                            ),
                            trailing: IconButton(
                                icon: Icon(
                                    Icons.delete_sweep,
                                    color: Colors.deepOrangeAccent
                                ),
                                onPressed: () {
                                    FirebaseFirestore.instance.collection('items').doc(snapshot.data.docs[index].id).de
lete();
                                },
                            ),
                        ),
                    onDismissed: (direction) {
                        FirebaseFirestore.instance.collection('items').doc(snapshot.data.docs[index].id).de
lete();
                    },
                );
            },
        );
    },
);

```

```
    }  
    );  
    },  
    ),  
    );  
}  
}
```