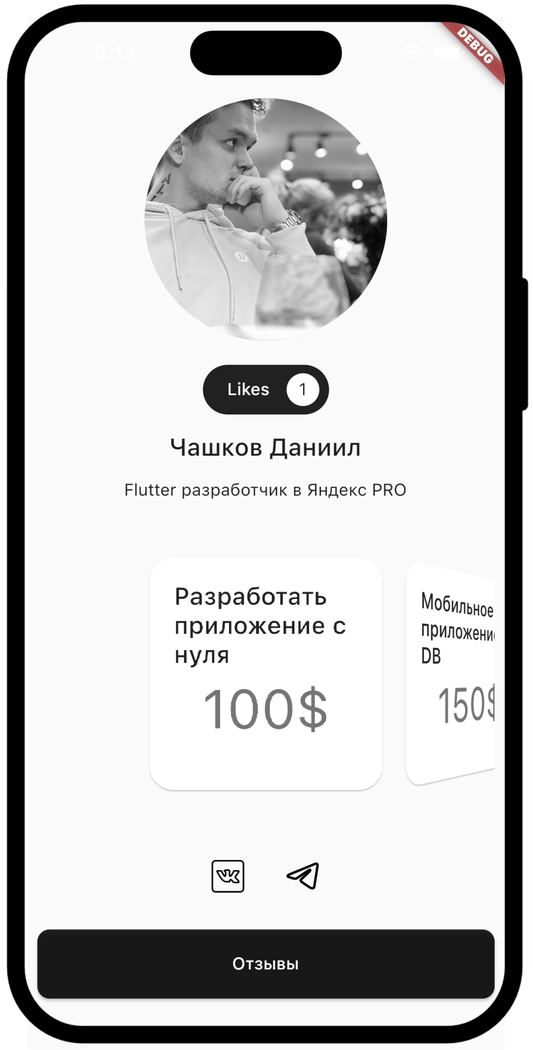
Яндекс

Виджеты

Flutter разработчик Яндекс Про
Чашков Даниил

Showcase App



Содержит все необходимые примеры:
Stateless, Stateful, Inherited, взаимодействие с основными виджетами

Содержание

01

Что такое виджет

Все, что Вы видите на экране - виджет

Виджет аватарки

Виджет лайков

Виджет иконки

Каталог виджетов

Неизменяемая конфигурация

- Описывает UI Flutter приложения
- Именно с виджетами мы работаем большую часть времени
- Сам виджет изменить нельзя, он помечен аннотацией @immutable и имеет константный конструктор
- Можно лишь удалить старый виджет и создать новый с другой конфигурацией (конфигурация - параметры конструктора виджета)
- Виджеты - древовидная структура виджетов (но не дерево)

```
@immutable
abstract class Widget extends DiagnosticableTree {
  const Widget({this.key});

  final Key? key;

  Element createElement();

  static bool canUpdate(Widget oldWidget, Widget newWidget) {
    return oldWidget.runtimeType == newWidget.runtimeType &&
        oldWidget.key == newWidget.key;
  }
}
```

Вопросы

А теперь поехали рассмотрим виджеты

01

StatelessWidget
Виджет без состояния

StatelessWidget

- Самый простой в использовании виджет
- Не имеет состояния
- Может перерисовывать, если родитель скажет ему об этом (т.е. конфигурация родительского виджета поменялась или мы зависим от InheritedWidget, который как-то изменился)
- Наполняет UI данными, которые ему передали

```
abstract class StatelessWidget extends Widget {
  const StatelessWidget({super.key});

  @override
  StatelessElement createElement() => StatelessElement(this);

  Widget build(BuildContext context);
}
```

Основные виджеты

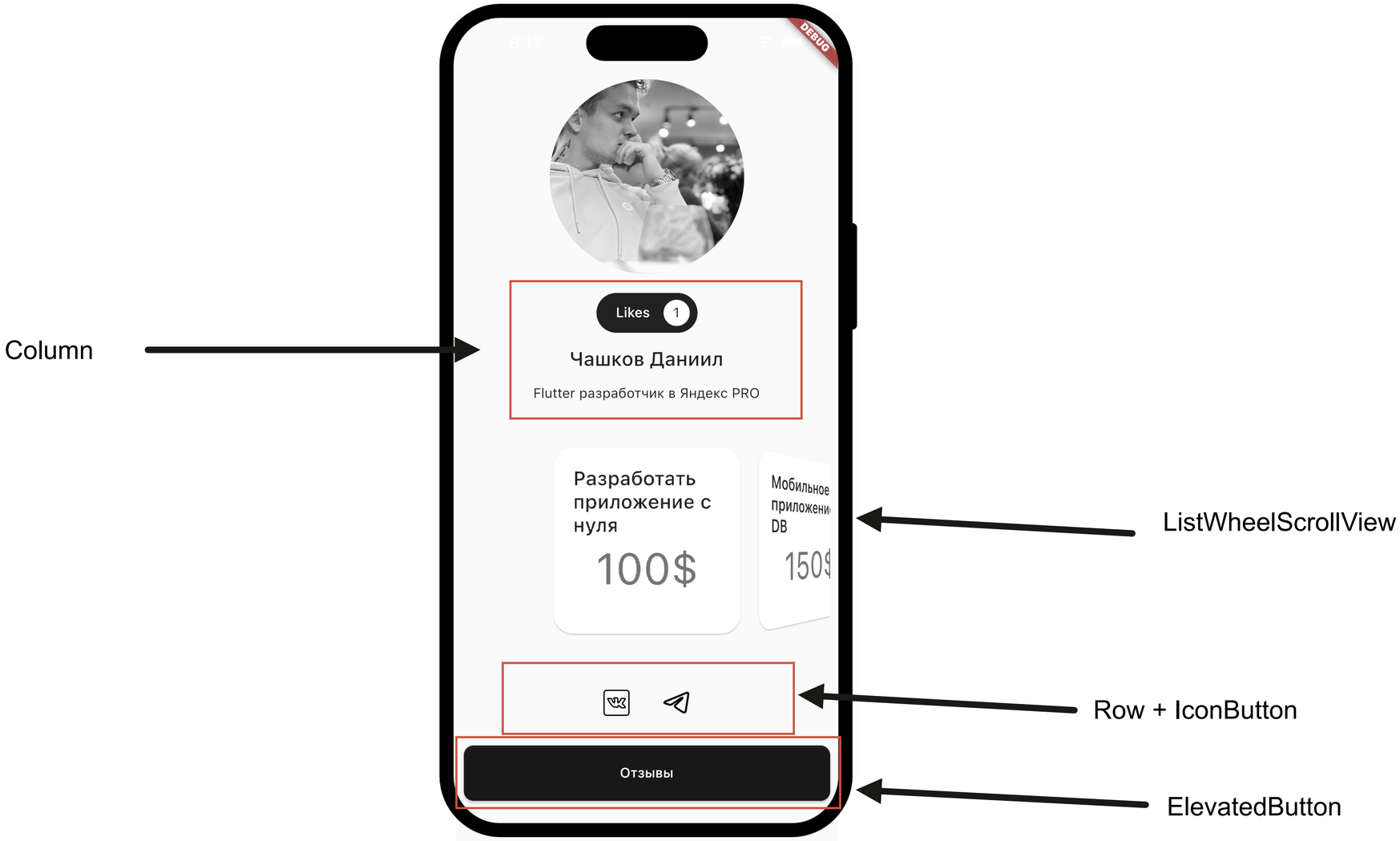
Single/content виджеты

- *Container* - виджет и рисует, и добавляет отступы, тени, цвет фона и мн. др.
- *Text* - отображает текст
- *SizedBox* - задает размер дочернему виджету
- *Padding* - задает внутренний отступ для ребенка
- *ElevatedButton, TextButton, OutlinedButton* - кнопки
- *Expanded* - виджет занимает все доступное пространство
- *Spacer* - виджет занимает все доступное пространство

Multichild виджеты

- *Column* - виджеты расположены друг за другом **вертикально**
- *Row* - виджеты расположены друг за другом **горизонтально**
- *Stack* - виджеты расположены один НАД другим (как бы слоями)
- *ListView* - виджеты расположены друг за другом **вертикально или горизонтально** и их может скролить

Основные виджеты

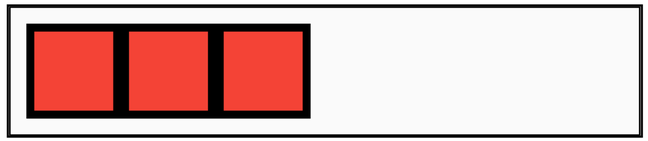


Пример на нашем showcase app

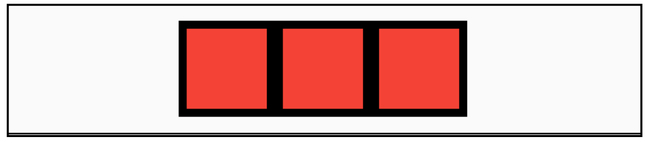
Расположение внутри Row, Column

MainAxisAlignment

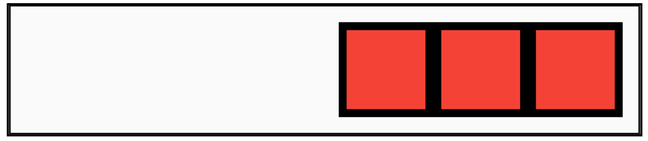
start



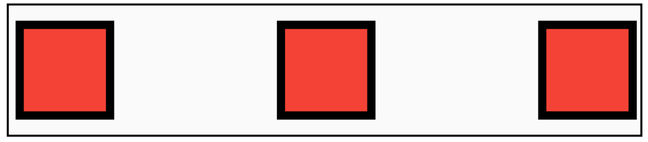
center



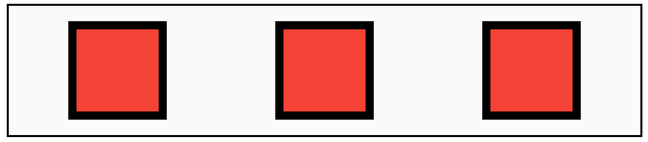
end



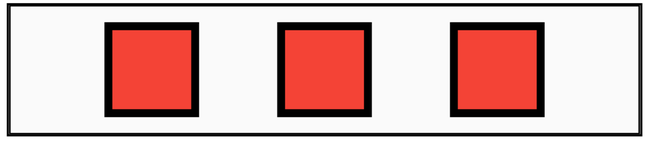
spaceBetween



spaceAround

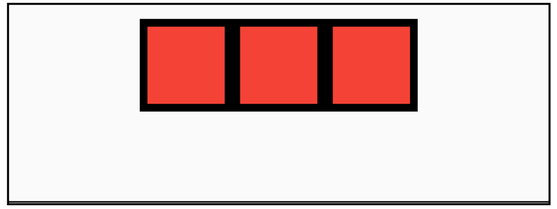


spaceEvenly

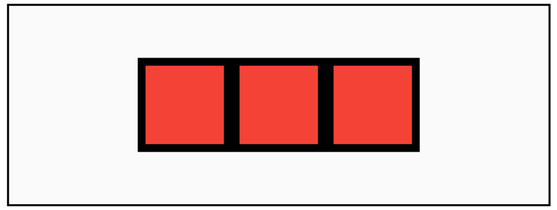


CrossAxisAlignment

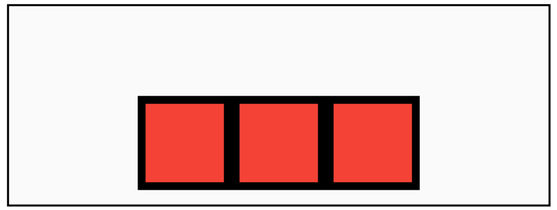
start



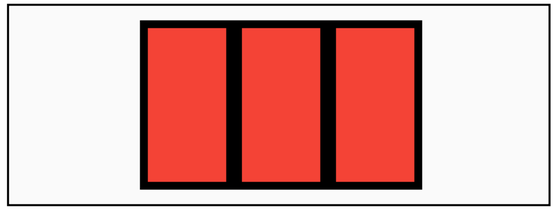
center



end



stretch



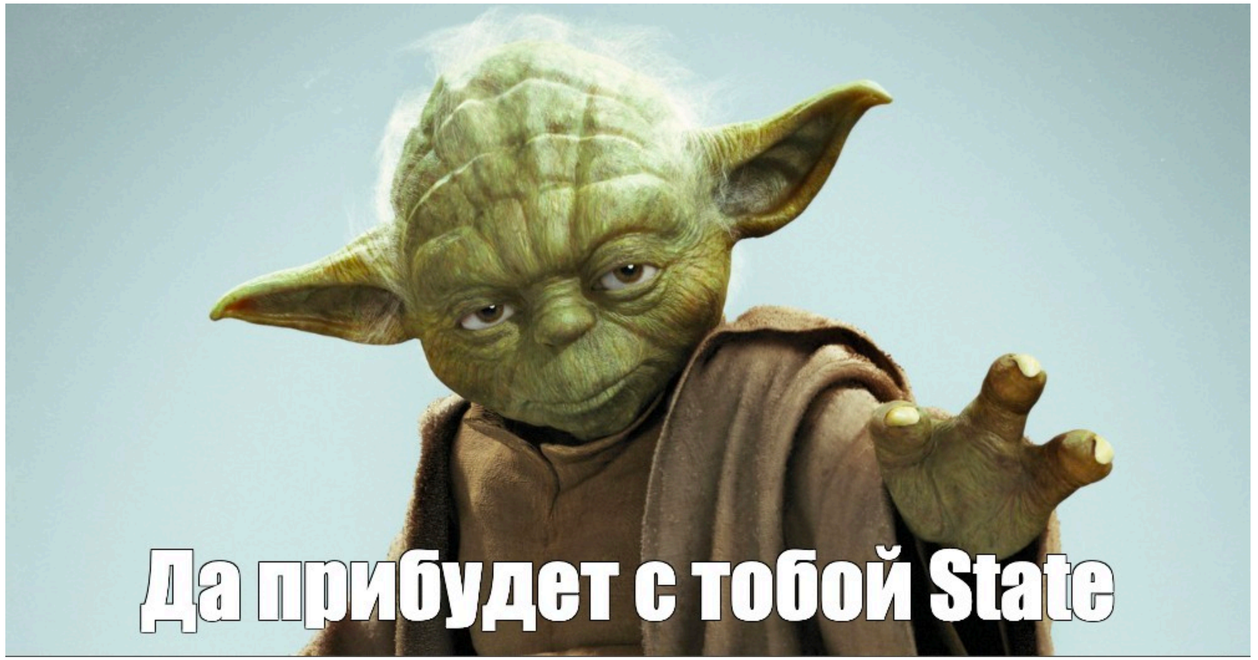
Вопросы

02

StatefulWidget
Виджет с состоянием

StatefulWidget

- Описывает UI Flutter приложения
- Содержит стейт и жизненные циклы (стейт - отдельный класс)
- Сам по себе виджет так же неизменяем, но не его стейт
- Мы сами может заставить виджет перерисоваться вызывая метод setState()



StatefulWidget - жизненный цикл

- **initState —** вызываться только один раз при инициализации стейта
- **didChangeDependencies —** единожды после инициализации и далее при изменении виджетов, на которые подписались, т.е. InheritedWidget
- **didUpdateWidget —** каждый раз при обновлении виджета
- **build —** каждый раз при перерисовке
- **dispose —** при удалении из дерева

- **setState —** вызываем для перерисовки

Пример на нашем showcase app

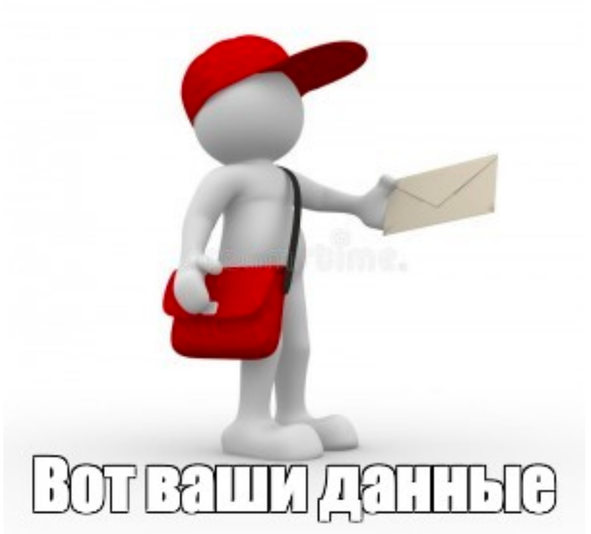
Вопросы

03

InheritedWidget
Виджет с данными

InheritedWidget - доступ к данным

- Решает проблему множественной передачи данных через конструкторы
- Получение данных, только при наличии контекста
- Может быть доступен для виджетов ниже по дереву с помощью *dependOnInheritedWidgetOfExactType* или *findAncestorWidgetOfExactType*
- Виджеты, которые использовали метод *dependOnInheritedWidgetOfExactType* для получения данных, могут быть перерисованы если изменилась конфигурация виджет InheritedWidget
- При изменении InheritedWidget будут перерисованы все виджеты, которые подписались на него и **не** const виджеты



```
class TitleInheritedWidget extends InheritedWidget {
  final String title;
  const TitleInheritedWidget(this.title, super.child);

  @override
  bool updateShouldNotify(covariant TitleInheritedWidget oldWidget) {
    return title != oldWidget.title;
  }

  static TitleInheritedWidget? of(BuildContext context) {
    final widget = context.
        dependOnInheritedWidgetOfExactType<TitleInheritedWidget>();
    return widget;
  }
}
...
class SomeWidget extends StatelessWidget {
   const SomeWidget();

   Widget build(context) {
     return TitleInheritedWidget("THIS IS TITLE", FirstWidget());
   }
}
...
class HundredthWidget extends StatelessWidget {
   const HundredthWidget();

   Widget build(context) {
     final title = TitleInheritedWidget.of(context).title;
     return Text(title);
   }
}
```

```
class FirstWidget extends StatelessWidget {
   final String title;
   const FirstWidget(this.title);

   Widget build(context) {
     return SecondWidget(title);
   }
}
class SecondWidget extends StatelessWidget {
   final String title;
   const SecondWidget(this.title);

   Widget build(context) {
     return ThirdWidget(title);
   }
}
...
class HundredthWidget extends StatelessWidget {
   final String title;
   const HundredthWidget(this.title);

   Widget build(context) {
     return Text(title);
   }
}
```

Пример на нашем showcase app

Вопросы

04

InheritedNotifier
Виджет для автоматической перестройки

InheritedNotifier - автоперестройка

- Позволяет обновить свои данные в отличие от *InheritedWidget*
- В остальном решает те же проблемы, что и *InheritedWidget*
- Вам не нужно указывать что именно обновилось в виджете, достаточно обновить данные
- Так же при обновлении данных будут перерисовываться только те виджет, которые подписались на него
- *findAncestorWidgetOfExactType* - лучше использовать для вызова методов InheritedNotifier
- *dependOnInheritedWidgetOfExactType* - лучше использовать для того, чтобы подписать виджет на обновление данных

Пример на нашем showcase app

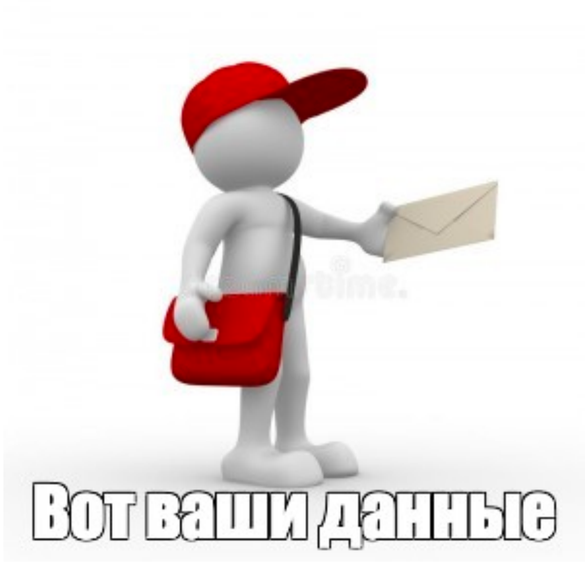
Вопросы

05

InheritedModel
Виджет для точечной перестройки

InheritedModel - точечная перелицовка

- Решает те, же самые проблемы, что и InheritedWidget
- Но есть отличие, виджет могут подписываться не на все обновления данных, а только на конкретное обновление определенного типа данных с помощью параметра aspect



Советы

- Используйте перерисовку точечно. Только там, где нужно (не на весь экран, а только на конкретные виджеты)
- Старайтесь создавать виджеты с *const* конструктором
- Не создавайте функции внутри виджетов, которые возвращают другие виджеты, лучше создавать отдельные виджеты
- Не стоит опираться на метод build для написания какой-то логики, так как мы не всегда можем контролировать build
- Помните про опасность InheritedWidget

Вопросы

Яндекс

Контакты

Даниил Чашков

Telegram:

@way_of_anaconda