Лабораторная № 6 Navigation. Method Chanel

1. Добавить к предыдущей лабораторной работе логику переходов между экранами используя Navigator, также создать переходы с передачей данных между экранами.

2. Добавить на один из экранов widget PageViewer продемонстрировать его работу

3. Создать Platform Method Channel в зависимости от вашего физического устройства или эмулятора (Android или iOS), метод возвращать информацию Flutter (обычно недоступную как пример: уровень заряда батареи, открытие набора телефонного номера, проверка разрешений и т.д.) Выполняется по вариантам.

|  |  |
| --- | --- |
| Вариант | Задание |
| 1 | Получить текущий ориентацию устройства |
| 2 | Получить тип сети к которую использует устройство (Wifi/Мобильный интернет) |
| 3 | Получить информацию заражается ли устройство |
| 4 | Получить название производителя устройства |
| 5 | Получить ip адрес |
| 6 | Вывести показания акселерометра |
| 7 | Получить текущую геопозицию. |
| 8 | Получить текущую локализацию устройства |
| 9 | Получить часовой пояс устройства |
| 10 | Получить статус Bluetooth |
| 11 | Получить SSID (имя) подключенной сети wifi |
| 12 | Получить информацию включен ли GPS |
| 13 | Получение текущей загрузки процессора |
| 14 | Получить информацию с датчика освещённости |
| 15 | Получить данные с гироскопа |

3. Создать Platform Method Channel передающий данные для запуска встроенного приложения. Все данные передаются из Flutter

|  |  |
| --- | --- |
| Вариант | Задание |
| 1, 7, 14 | Запуск приложение почты с заполнением темы сообщения |
| 2, 8, 15 | Запуск приложения карты с переданными координатами |
| 3, 6, 12 | Запуск отправки сообщения с заполненным телом сообщения |
| 4, 10, 9 | Запуск браузера с переданным URL |
| 5, 11, 13 | Запуск приложения для установки будильника с заданным временем |

4 На основе документации также добавить два Platform Method Channel для IOS и Android для получения заряда батареи. Показать разницу.

Вопросы по лабораторной работе:

1. Компонент Navigator, какие основные методы взаимодействия?

2. Виды route?

3. Что такое Platform Method Channel?

4. Как производится валидация вызываемого method channel?

1. Компонент Navigator:

Основные методы:

push: добавляет новый маршрут на вершину стека.

pop: удаляет текущий маршрут и возвращает на предыдущий.

pushReplacement: заменяет текущий маршрут новым.

pushNamed: навигация с использованием именованных маршрутов.

popUntil: удаляет маршруты до указанного условия.

2. Виды route:

MaterialPageRoute: стандартный маршрут с анимацией переходов для платформы Android.

CupertinoPageRoute: стиль переходов в iOS.

NamedRoute: маршруты с использованием строковых имен для навигации.

CustomRoute: позволяет настраивать анимацию переходов между экранами.

Platform Method Channel:

3. Это механизм взаимодействия между кодом на Flutter (Dart) и платформенным кодом (напр., Java/Kotlin для Android или Swift/Objective-C для iOS) для выполнения платформо-специфичных задач.

**Валидация вызываемого method channel**:

* **Проверка существования метода**: важно убедиться, что метод, который вы хотите вызвать на стороне платформы, существует и правильно зарегистрирован.
* **Проверка данных**: передаваемые данные должны быть корректно сериализованы и десериализованы, чтобы избежать ошибок типа или формата.
* **Обработка ошибок**: вызовы через MethodChannel могут завершиться с ошибкой. В Dart-коде можно использовать блоки try-catch, чтобы перехватывать возможные исключения. Со стороны нативного кода также можно возвращать ошибки или статус выполнения операции.
* **Асинхронное выполнение**: методы могут выполняться асинхронно, поэтому важно корректно обрабатывать возвращаемые значения (например, Future в Dart) и следить за тем, что метод завершился успешно.