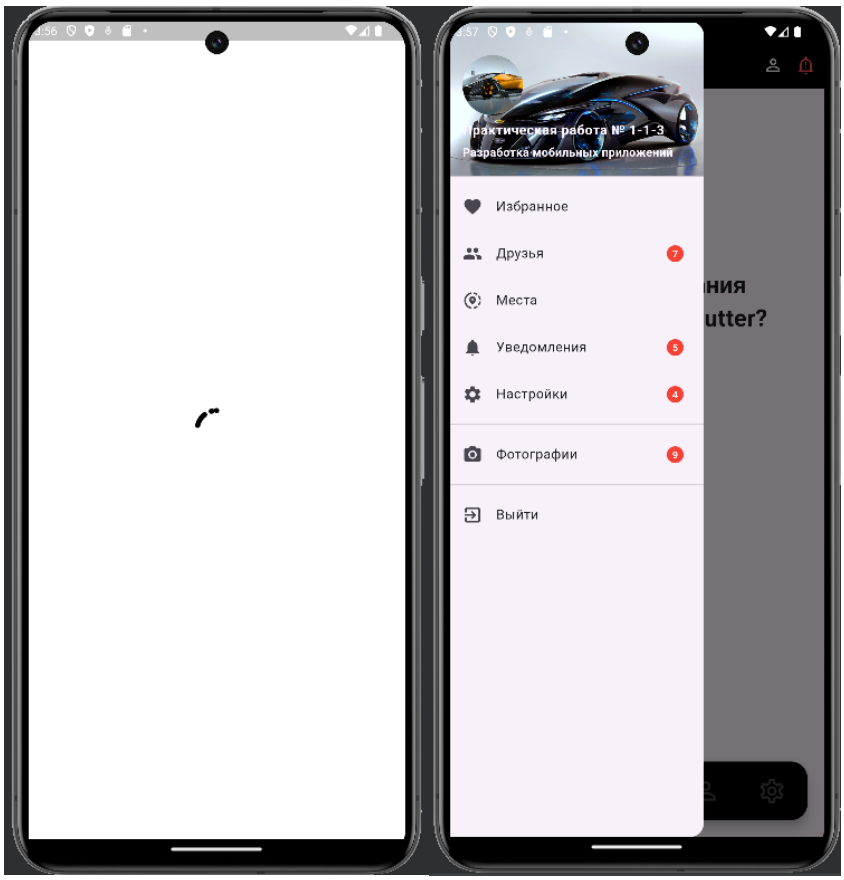
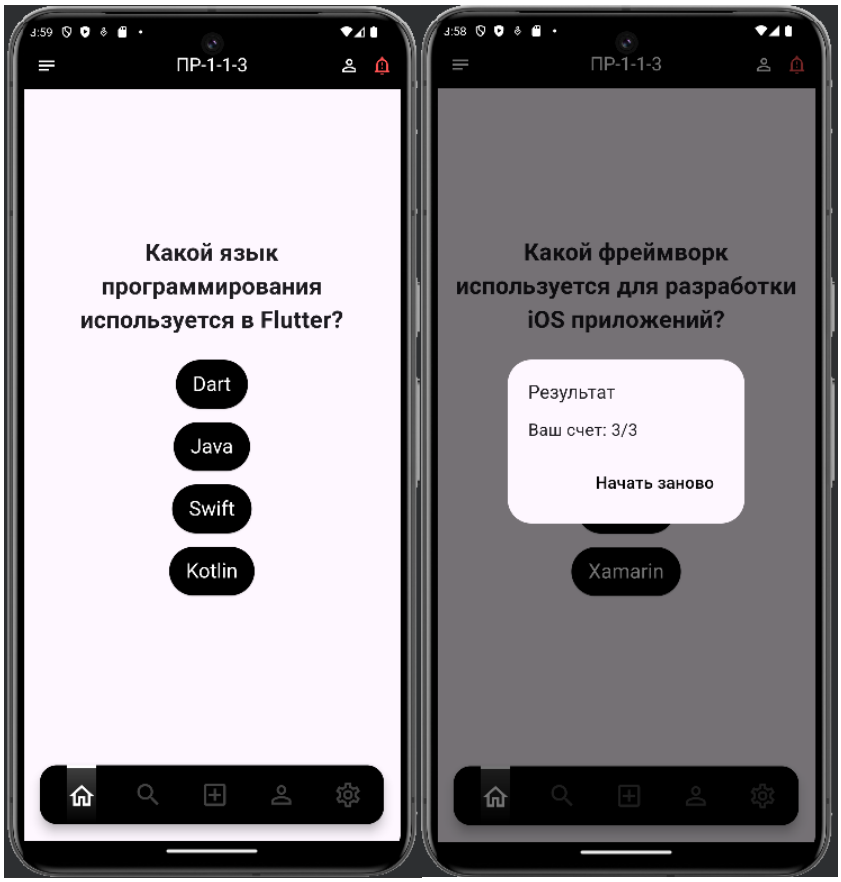
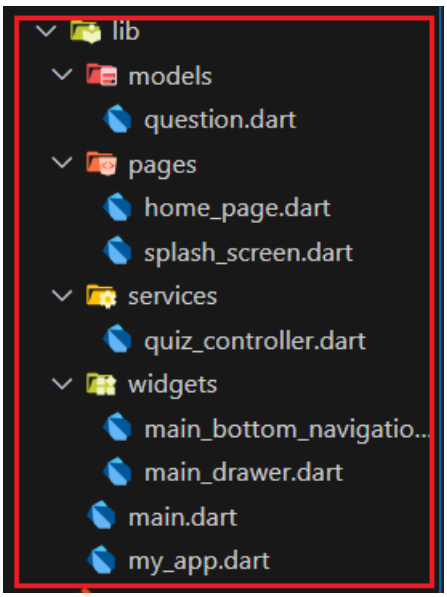
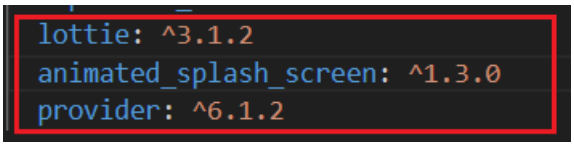
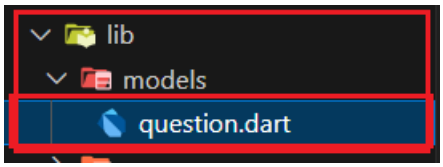

Практическая работа № 1-1-3
Разработка мобильных приложений
Избранное
Друзья
7
Места
Уведомления
5
Настройки
4
Фотографии
9
Выйти


ПР-1-1-3
Какой язык программирования используется в Flutter?
Dart
Java
Swift
Kotlin
ПР-1-1-3
Какой фреймворк используется для разработки iOS приложений?
Результат
Ваш счет: 3/3
Начать заново
Xamarin



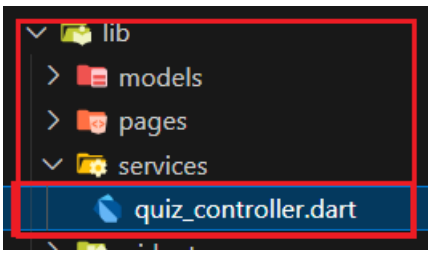





```dart
class Question {
  final String text;
  final List<String> options;
  final int correctAnswerIndex;

  Question({
    required this.text,
    required this.options,
    required this.correctAnswerIndex,
  });
}

final List<Question> questions = [
  Question(
    text: 'Какой язык программирования используется в Flutter?',
    options: ['Dart', 'Java', 'Swift', 'Kotlin'],
    correctAnswerIndex: 0,
  ),
  Question(
    text: 'Какая компания разработала Flutter?',
    options: ['Google', 'Apple', 'Microsoft', 'Facebook'],
    correctAnswerIndex: 0,
  ),
  Question(
    text: 'Какой фреймворк используется для разработки iOS приложений?',
    options: ['Flutter', 'React Native', 'SwiftUI', 'Xamarin'],
    correctAnswerIndex: 2,
  ),
];
```




```dart
class QuizController extends ChangeNotifier {
  final List<Question> questions;
  int currentQuestionIndex = 0;
  int score = 0;

  QuizController(this.questions);

  void nextQuestion(int selectedAnswerIndex, BuildContext context) {
    if (questions[currentQuestionIndex].correctAnswerIndex ==
        selectedAnswerIndex) {
      score++;
    }
    if (currentQuestionIndex < questions.length - 1) {
      currentQuestionIndex++;
    } else {
      showResultDialog(context);
    }
    notifyListeners();
  }

  void resetQuiz() {
    currentQuestionIndex = 0;
    score = 0;
    notifyListeners();
  }

  bool get isQuizEnd => currentQuestionIndex == questions.length - 1;
```

```dart
void showResultDialog(BuildContext context) {
  showDialog(
    context: context,
    builder: (context) => AlertDialog(
      title: const Text(
        'Результат',
        style: TextStyle(
          fontSize: 22,
        ), // TextStyle
      ), // Text
      content: Text(
        'Ваш счет: $score/${questions.length}',
        style: const TextStyle(
          fontSize: 20,
        ), // TextStyle
      ), // Text
      actions: [
        TextButton(
          onPressed: () {
            resetQuiz();
            Navigator.of(context).pop();
          },
          child: const Text(
            'Начать заново',
            style: TextStyle(
              color: Colors.black,
              fontSize: 20,
            ), // TextStyle
          ), // Text
        ), // TextButton
      ],
    ), // AlertDialog
  );
 }
}
```

QuizController

QuizController — это класс, который управляет состоянием теста. Он наследуется от *ChangeNotifier*, что позволяет ему уведомлять слушателей об изменениях состояния.

Поля:

final List<Question> questions: Список вопросов викторины.

int currentQuestionIndex = 0: Индекс текущего вопроса.

int score = 0: Счет пользователя.

Конструктор:

QuizController(this.questions): Принимает список вопросов и инициализирует их.

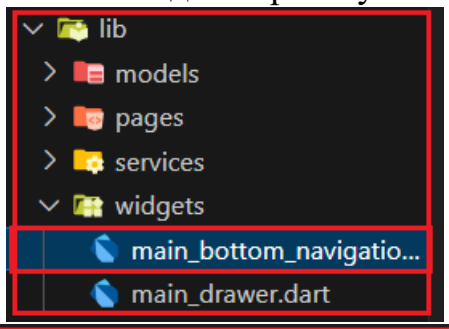
Методы:

void nextQuestion(int selectedAnswerIndex, BuildContext context): Обрабатывает выбор пользователя. Если ответ правильный, увеличивает счет. Если текущий вопрос последний, вызывает *showResultDialog*.

void resetQuiz(): Сбрасывает состояние теста, устанавливая индекс текущего вопроса в 0 и счет в 0.

bool get isQuizEnd => currentQuestionIndex == questions.length - 1: Возвращает true, если текущий вопрос является последним.

void showResultDialog(BuildContext context): Показывает диалоговое окно с результатами викторины и кнопкой для перезапуска.




```dart
class MainBottomNavigationBar extends StatefulWidget {
  const MainBottomNavigationBar({super.key});

  @override
  State<MainBottomNavigationBar> createState() => _MainBottomNavigationBarState();
}

class _MainBottomNavigationBarState extends State<MainBottomNavigationBar> {
  int iconSelectedIndex = 0;
  List<IconData> icons = [
    Icons.home_outlined,
    Icons.search,
    Icons.add_box_outlined,
    Icons.person_outline,
    Icons.settings_outlined,
  ];
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding: const EdgeInsets.all(20),
      child: Material(
        elevation: 10,
        borderRadius: BorderRadius.circular(20),
        color: Colors.black,
        // ignore: sized_box_for_whitespace
        child: Container(
          height: 70,
          width: double.infinity,
          child: ListView.builder(
            scrollDirection: Axis.horizontal,
            itemCount: icons.length,
            padding: const EdgeInsets.symmetric(horizontal: 10),
            itemBuilder: (context, index) => Padding(
              padding: const EdgeInsets.symmetric(horizontal: 22),
              child: GestureDetector(
```

```
        onTap: () {
          setState(() {
            iconSelectedIndex = index;
          });
        },
        child: AnimatedContainer(
          duration: const Duration(milliseconds: 250),
          width: 35,
          decoration: BoxDecoration(
            border: index == iconSelectedIndex
                ? const Border(
                top: BorderSide(width: 3, color: Colors.white)) // Border
                : null,
            gradient: index == iconSelectedIndex
                ? LinearGradient(
              begin: Alignment.topCenter,
              end: Alignment.bottomCenter,
              colors: [
                Colors.grey.shade800,
                Colors.black,
              ],
            ) // LinearGradient
                : null,
          ), // BoxDecoration
          child: Icon(
            icons[index],
            size: 35,
            color: index == iconSelectedIndex
                ? Colors.white
                : Colors.grey.shade800,
          ), // Icon
        ), // AnimatedContainer
      ), // GestureDetector
    ), // Padding
  ), // ListView.builder
```

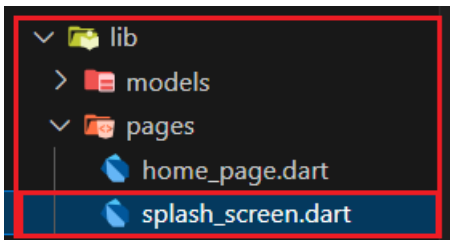
```dart
class MainDrawer extends StatelessWidget {
  const MainDrawer({super.key});

  @override
  Widget build(BuildContext context) {
    return Drawer(
      child: ListView(
        padding: EdgeInsets.zero,
        children: <Widget>[
          UserAccountsDrawerHeader(
            accountName: const Text(
              'Практическая работа № 1-1-3',
              style: TextStyle(
                fontWeight: FontWeight.bold,
              ), // TextStyle
            ), // Text
            accountEmail: const Text(
              'Разработка мобильных приложений',
              style: TextStyle(
                fontWeight: FontWeight.bold,
              ), // TextStyle
            ), // Text
            currentAccountPicture: CircleAvatar(
              child: ClipOval(
                child: Image.network(
                  'https://4-x-4.ru/wp-content/uploads/2023/05/1295566be0f5d1ba3ff144fe6c631515.jpeg',
                  height: 90,
                  width: 90,
                  fit: BoxFit.cover,
                ), // Image.network
              ), // ClipOval
            ), // CircleAvatar
            decoration: const BoxDecoration(
              color: Colors.redAccent,
              image: DecorationImage(
                image: NetworkImage(
                    'https://img.goodfon.ru/original/4096x2734/d/6d/2015-chevrolet-fnr-concept.jpg'), // NetworkImage
                fit: BoxFit.cover,
              ), // DecorationImage
            ), // BoxDecoration
          ), // UserAccountsDrawerHeader
          ListTile(
            leading: const Icon(Icons.favorite),
            title: const Text('Избранное'),
            onTap: () {},
          ), // ListTile
          ListTile(
            leading: const Icon(Icons.people),
            title: const Text('Друзья'),
            onTap: () {},
            trailing: ClipOval(
              child: Container(
                color: Colors.red,
                width: 20,
                height: 20,
                child: const Center(
                  child: Text(
                    '7',
                    style: TextStyle(
                      color: Colors.white,
                      fontSize: 12,
                    ), // TextStyle
                  ), // Text
                ), // Center
              ), // Container
            ), // ClipOval
          ), // ListTile
```

```
        ListTile(
          leading: const Icon(Icons.share_location),
          title: const Text('Места'),
          onTap: () {},
        ), // ListTile
        ListTile(
          leading: const Icon(Icons.notifications),
          title: const Text('Уведомления'),
          onTap: () {},
          trailing: ClipOval(
            child: Container(
              color: Colors.red,
              width: 20,
              height: 20,
              child: const Center(
                child: Text(
                  '5',
                  style: TextStyle(
                    color: Colors.white,
                    fontSize: 12,
                  ), // TextStyle
                ), // Text
              ), // Center
            ), // Container
          ), // ClipOval
        ), // ListTile
        ListTile(
          leading: const Icon(Icons.settings),
          title: const Text('Настройки'),
          onTap: () {},
          trailing: ClipOval(
            child: Container(
```

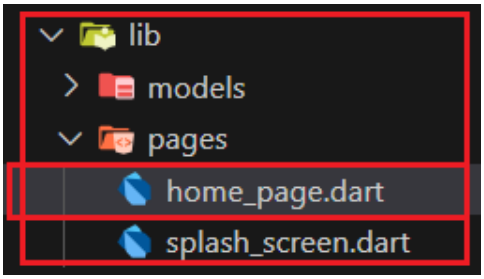
```
            color: Colors.red,
            width: 20,
            height: 20,
            child: const Center(
              child: Text(
                '4',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 12,
                ), // TextStyle
              ), // Text
            ), // Center
          ), // Container
        ), // ClipOval
      ), // ListTile
      const Divider(),
      ListTile(
        leading: const Icon(Icons.photo_camera),
        title: const Text('Фотографии'),
        onTap: () {},
        trailing: ClipOval(
          child: Container(
            color: Colors.red,
            width: 20,
            height: 20,
            child: const Center(
              child: Text(
                '9',
                style: TextStyle(
                  color: Colors.white,
                  fontSize: 12,
                ), // TextStyle
              ), // Text
            ), // Center
          ), // Container
```

```
        const Divider(),
        ListTile(
          leading: const Icon(Icons.exit_to_app),
          title: const Text('Выйти'),
          onTap: () {},
        ), // ListTile
      ], // <Widget>[]
    ), // ListView
  ); // Drawer
}
```




```dart
class SplashScreen extends StatelessWidget {
  const SplashScreen({super.key});

  @override
  Widget build(BuildContext context) {
    return AnimatedSplashScreen(
      splash: Center(
        child: Lottie.asset(
          'assets/images/main_logo.json',
        ),
      ), // Center
      nextScreen: const HomePage(),
      duration: 3500,
      backgroundColor: Colors.white,
    ); // AnimatedSplashScreen
  }
}
```



```dart
class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}

class _HomePageState extends State<HomePage> {
  @override
  Widget build(BuildContext context) {
    final controller = context.watch<QuizController>();
    return Scaffold(
      appBar: AppBar(
        elevation: 0,
        leading: Builder(
          builder: (context) => IconButton(
              onPressed: () {
                Scaffold.of(context).openDrawer();
              },
              icon: const Icon(
                Icons.notes,
                color: Colors.white,
              )), // Icon // IconButton
        ), // Builder
        backgroundColor: Colors.black,
        title: const Text(
          'ПР-1-1-3',
          style: TextStyle(color: Colors.white),
        ), // Text
        centerTitle: true,
        actions: const [
          Icon(
            Icons.person_outline,
            color: Colors.white,
          ), // Icon
          SizedBox(width: 15),
```

```
          Icon(
            Icons.notification_important_outlined,
            color: Colors.redAccent,
          ), // Icon
          SizedBox(width: 10),
        ],
      ), // AppBar
      drawer: const MainDrawer(),
      body: Center(
        child: Padding(
          padding: const EdgeInsets.all(16.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(
                controller.questions[controller.currentQuestionIndex].text,
                style:
                    const TextStyle(fontSize: 28, fontWeight: FontWeight.bold),
                textAlign: TextAlign.center,
              ), // Text
              const SizedBox(height: 20),
              ...controller.questions[controller.currentQuestionIndex].options
                  .map((option) {
                int index = controller
                    .questions[controller.currentQuestionIndex].options
                    .indexOf(option);
                return Padding(
                  padding: const EdgeInsets.symmetric(vertical: 8.0),
                  child: ElevatedButton(
                    onPressed: () => controller.nextQuestion(index, context),
                    style: ElevatedButton.styleFrom(
                      foregroundColor: Colors.white,
                      backgroundColor: Colors.black,
                      padding: const EdgeInsets.symmetric(
                          horizontal: 20, vertical: 15), // EdgeInsets.symmetric
                      textStyle: const TextStyle(
                        fontSize: 24,
                      ), // TextStyle
                    ),
                    child: Text(option),
                  ), // ElevatedButton
                ); // Padding
              }),
            ],
          ), // Column
        ), // Padding
      ), // Center
      bottomNavigationBar: const MainBottomNavigationBar(),
    ); // Scaffold
  }
```

Разметка body:

Center: центрирует все содержимое по центру экрана.

Padding: добавляет отступы вокруг содержимого для улучшения визуального восприятия.

Column: Контейнер, который располагает свои дочерние элементы вертикально.

Text: отображает текст текущего вопроса. Стиль текста увеличен для лучшей видимости.

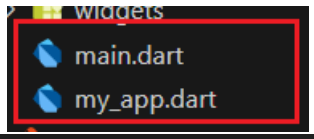
SizedBox(height: 20): пустой виджет, который добавляет вертикальный отступ между вопросом и вариантами ответов.

...: оператор распаковки списка, который позволяет вставить все элементы списка в родительский виджет.

Padding: добавляет отступы вокруг каждой кнопки с вариантом ответа.

ElevatedButton: Кнопка, которая отображает вариант ответа. При нажатии вызывает метод controller.nextQuestion(index, context), передавая индекс выбранного ответа и контекст для отображения диалогового окна с результатами.

SizedBox(height: 20): Пустой виджет, который добавляет вертикальный отступ в конце списка вариантов ответов.




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

  @override
  Widget build(BuildContext context) {
    return ChangeNotifierProvider(
      create: (context) => QuizController(questions),
      child: MaterialApp(
        debugShowCheckedModeBanner: false,
        title: 'Flutter PR Task - 1-1-3',
        theme: ThemeData(
          colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
          useMaterial3: true,
        ), // ThemeData
        home: const SplashScreen(),
      ), // MaterialApp
    ); // ChangeNotifierProvider
  }
}
```

```dart
Run | Debug | Profile
void main() {
  runApp(const MyApp());
}
```