

Документ подписан простой электронной подписью

Информация о документе:

ФИО: Хоружий Людмила Ивановна

Должность: Директор института экономики и управления АПК

Дата подписания: 24.08.2025 11:21:14

Уникальный программный ключ:

1e90b132d9b04dce67585160b015dddf2cb1e6a9



**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ –  
МСХА имени К.А. ТИМИРЯЗЕВА»**

**(ФГБОУ ВО РГАУ - МСХА имени К.А. Тимирязева)**

Институт экономики и управления АПК  
Кафедра статистики и кибернетики

УТВЕРЖДАЮ:  
Директор института  
экономики и управления АПК  
Л.И. Хоружий  
«28» августа 2025 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
Б1.В.17 Разработка мобильных приложений**

для подготовки бакалавров

ФГОС ВО

Направление: 09.03.02 «Информационные системы и технологии»

Направленность:

«Фуллстек разработка»

Курс 4

Семестр 7

Форма обучения: очная

Год начала подготовки: 2025

Москва, 2025

Разработчики:

Демичев В.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

Храмов Д.Э., ассистент

(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

Рецензент:

Вахрушева И.А., канд. пед. наук

(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, профессионального стандарта и учебного плана 2025 года начала подготовки.

Программа обсуждена на заседании кафедры статистики и кибернетики. Протокол №11 от «26» августа 2025 г.

И. о. зав. кафедрой Уколова А.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)

  
(подпись)

«26» августа 2025 г.

**Согласовано:**

Председатель учебно-методической  
комиссии института экономики и управления АПК

Гупалова Т.Н., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)

  
(подпись)

протокол №1 «28» августа 2025 г.

И. о. зав. выпускающей кафедрой  
статистики и кибернетики

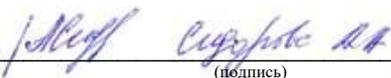
Уколова А.В., канд. экон. наук, доцент

(ФИО, ученая степень, ученое звание)

  
(подпись)

«28» августа 2025 г.

Заведующий отделом комплектования ЦНБ

  
(подпись)

## СОДЕРЖАНИЕ

<b>АННОТАЦИЯ .....</b>	<b>4</b>
<b>1. ЦЕЛЬ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>5</b>
<b>2. МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ .....</b>	<b>5</b>
<b>3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ.....</b>	<b>6</b>
<b>4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ .....</b>	<b>13</b>
4.1 РАСПРЕДЕЛЕНИЕ ТРУДОЁМКОСТИ ДИСЦИПЛИНЫ ПО ВИДАМ РАБОТ .....	13
ПО СЕМЕСТРАМ .....	13
4.2 СОДЕРЖАНИЕ ДИСЦИПЛИНЫ.....	13
4.3 ПРАКТИЧЕСКИЕ ЗАНЯТИЯ.....	15
<b>5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ .....</b>	<b>20</b>
<b>6. ТЕКУЩИЙ КОНТРОЛЬ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ .....</b>	<b>21</b>
6.1. ТИПОВЫЕ КОНТРОЛЬНЫЕ ЗАДАНИЯ ИЛИ ИНЫЕ МАТЕРИАЛЫ, НЕОБХОДИМЫЕ ДЛЯ ОЦЕНКИ ЗНАНИЙ, УМЕНИЙ И НАВЫКОВ И (ИЛИ) ОПЫТА ДЕЯТЕЛЬНОСТИ .....	21
6.2. ОПИСАНИЕ ПОКАЗАТЕЛЕЙ И КРИТЕРИЕВ КОНТРОЛЯ УСПЕВАЕМОСТИ, ОПИСАНИЕ ШКАЛ ОЦЕНИВАНИЯ.....	21
<b>Закладка не определена.</b>	
<b>7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....</b>	<b>28</b>
7.1 ОСНОВНАЯ ЛИТЕРАТУРА .....	28
7.2 ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА.....	<b>Ошибка! Закладка не определена.</b>
7.3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ, РЕКОМЕНДАЦИИ И ДРУГИЕ МАТЕРИАЛЫ К ЗАНЯТИЯМ..	<b>Ошибка! Закладка не определена.</b>
<b>8. ПЕРЕЧЕНЬ РЕСУРСОВ ИНФОРМАЦИОННО-ТЕЛЕКОММУНИКАЦИОННОЙ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫХ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ).....</b>	<b>Ошибка! Закладка не определена.</b>
<b>9. ПЕРЕЧЕНЬ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ И ИНФОРМАЦИОННЫХ СПРАВОЧНЫХ СИСТЕМ.....</b>	<b>28</b>
<b>10. ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ДИСЦИПЛИНЕ.....</b>	<b>30</b>
<b>11. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ СТУДЕНТАМ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ .....</b>	<b>32</b>
<b>12. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПРЕПОДАВАТЕЛЯМ ПО ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ.....</b>	<b>32</b>

## АННОТАЦИЯ

**рабочей программы учебной дисциплины Б1.В.17 «Разработка мобильных приложений» для подготовки бакалавров по направлению 09.03.02 Информационные системы и технологии, направленности «Фуллстек разработка»**

**Цель освоения дисциплины:** формирование у студентов профессиональных компетенций в области проектирования, разработки и развёртывания современных кроссплатформенных мобильных приложений с использованием актуальных фреймворков и архитектурных подходов на базе Flutter. В ходе изучения дисциплины студенты осваивают методы разработки как простых одноэкраных приложений, так и сложных многокомпонентных решений с управлением состоянием, навигацией, взаимодействием с backend и доступом к возможностям устройства. Также студенты приобретают навыки проектирования пользовательского интерфейса, интеграции с RESTful-сервисами, локального хранения данных, тестирования, а также подготовки мобильных приложений к публикации.

**Место дисциплины в учебном плане:** дисциплина включена в часть, формируемую участниками образовательных отношений дисциплин учебного плана по направлению подготовки 09.03.02 Информационные системы и технологии.

**Требования к результатам освоения дисциплины:** в результате освоения дисциплины формируются следующие компетенции (индикаторы): ПКос-3(ПКос-3.1, ПКос-3.2, ПКос-3.3), ПКос-5(ПКос-5.1, ПКос-5.2, ПКос-5.3), ПКос-7(ПКос-7.1, ПКос-7.2, ПКос-7.3).

**Краткое содержание дисциплины:**

Экосистема кроссплатформенной мобильной разработки: сравнение Flutter, React Native и .NET MAUI. Основы языка Dart: типизация, асинхронность, потоки, null safety. Архитектура Flutter: виджеты, дерево элементов, жизненный цикл, принцип реактивного UI. Структура проекта: папки, assets, зависимости, конфигурация. Виджеты и компоновка: StatelessWidget и StatefulWidget, Material и Cupertino, Row/Column, Stack, ListView, кастомные виджеты. Навигация и маршрутизация: именованные роуты, передача данных, возврат результата, BottomNavigationBar, Drawer. Управление состоянием: setState, Provider, Bloc — сравнение и применение. Работа с backend: HTTP-запросы (http, dio), обработка ошибок, аутентификация, работа с токенами. Локальное хранение данных: shared\_preferences, Hive, кэширование. Доступ к возможностям устройства: камера, геолокация, уведомления, файлы, через пакеты pub.dev. Тестирование: unit-тесты, widget-тесты, интеграционные тесты. Подготовка к публикации: настройка иконки, splash screen, сборка release-сборки (APK, IPA), описание для магазинов приложений.

**Общая трудоемкость дисциплины: 144 / 4 (часы/зач. ед.)**

**Промежуточный контроль: экзамен**

## **1. Цель освоения дисциплины**

Целью освоения дисциплины «Разработка мобильных приложений» является овладение студентами знаний архитектурных принципов построения кроссплатформенных мобильных приложений, а также особенностей реактивного UI, основных подходов к проектированию компонентной архитектуры, управления состоянием и взаимодействия с внешними сервисами. Студент должен знать принципы работы фреймворка Flutter, особенности языка Dart (включая асинхронность, потоки и null safety), механизмы компоновки виджетов, навигации и маршрутизации; понимать подходы к управлению состоянием (setState, Provider, Bloc), особенности локального хранения данных, интеграции с RESTful-сервисами, работы с возможностями устройства (камера, геолокация, уведомления), а также практики подготовки приложений к публикации в магазинах приложений. По окончании изучения дисциплины студент должен уметь проектировать структуру мобильного приложения в соответствии с поставленной задачей, создавать многоэкранные интерфейсы с навигацией, реализовывать обработку пользовательского ввода, интегрировать работу с backend-сервисами и локальным хранилищем, управлять состоянием приложения, обрабатывать доступ к аппаратным возможностям устройства, а также настраивать ресурсы и готовить приложение к публикации. Также по окончании изучения дисциплины студент должен владеть навыками разработки полнофункциональных кроссплатформенных мобильных приложений на Flutter с использованием декларативного подхода к UI, навыками применения современных паттернов управления состоянием (Provider, Bloc), методами тестирования (unit, widget, интеграционное), техниками оптимизации производительности и отзывчивости интерфейса, а также практиками документирования архитектуры приложения и оформления технических решений в соответствии с профессиональными стандартами мобильной разработки.

## **2. Место дисциплины в учебном процессе**

Дисциплина «Разработка мобильных приложений» включена в часть, формируемую участниками образовательных отношений дисциплин учебного плана. Дисциплина «Разработка мобильных приложений» реализуется в соответствии с требованиями ФГОС ВО, ОПОП ВО и Учебного плана по направлению 09.03.02 Информационные системы и технологии.

Дисциплина «Разработка мобильных приложений» изучается на четвертом курсе образовательного цикла.

Предшествующими курсами, включенными в учебный план, на которых непосредственно базируются дисциплина «Разработка мобильных приложений», являются «Алгоритмизация и программирование», «Теория информации», «Разработка профессиональных приложений».

Особенностью дисциплины является фокус на практико-ориентированном освоении современного стека кроссплатформенной мобильной разработки на основе Flutter, с акцентом на проектирование архитектуры приложений, управление состоянием, интеграцию с backend-сервисами, ис-

пользование возможностей устройства и подготовку приложений к публикации в магазинах приложений.

Рабочая программа дисциплины «Разработка мобильных приложений» для инвалидов и лиц с ограниченными возможностями здоровья разрабатывается индивидуально с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся.

### **3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы**

Образовательные результаты освоения дисциплины обучающимся, представлены в таблице 1.

Таблица 1

## Требования к результатам освоения учебной дисциплины «Разработка мобильных приложений»

№ п/п	Код компетенции	Содержание компетенции (или её части)	Индикаторы компетенций	В результате изучения учебной дисциплины обучающиеся должны:		
				знать	уметь	владеть
1.	ПКос-3	Способен проектировать и разрабатывать информационные ресурсы с использованием полного стека технологий	ПКос-3.1	Студент должен знать: архитектурные принципы кроссплатформенной мобильной разработки, реактивный UI, компонентный подход, разделение логики и представления; особенности языка Dart – типизацию, асинхронность, потоки, null safety; структуру проекта Flutter, назначение ключевых файлов (pubspec.yaml, main.dart), принципы работы виджетов, навигации и управления состоянием; типовые шаблоны и пакеты экосистемы Flutter для работы с API, локальным хранилищем, устройством, а также подходы к организации кода.		
			ПКос-3.2		Студент должен уметь: проектировать структуру мобильного приложения в соответствии с поставленной задачей, выделять экраны, определять потоки навига-	

					<p>ции, проектировать форму ввода и отображения данных;</p> <p>использовать типовые решения: многоэкранная навигация через Navigator или go_router, управление состоянием через Provider или Bloc, интеграция с REST API;</p> <p>применять пакеты из pub.dev для реализации доступа к камере, геолокации, уведомлениям;</p> <p>организовывать проект в соответствии с best practices: модульность, повторное использование виджетов, разделение ответственностей.</p>	
			ПКос-3.3			<p>Студент должен владеть:</p> <p>навыками проектирования и разработки полноценных кроссплатформенных мобильных приложений;</p> <p>практиками архитектурного проектирования, выбор паттерна управления состоянием под задачу, организация слоёв приложения, работа с асинхронными данными;</p> <p>навыками интеграции внешних сервисов, аутентификация, загрузка/сохранение данных, фоновые операции;</p>

						техниками подготовки приложения к распространению: настройка иконки, splash screen, локализация, сборка release-сборки, оформление метаданных для магазинов.
2.	ПКос-5	Способен осуществлять методическое сопровождение испытаний системы	ПКос-5.1	Студент должен знать: основные критерии оценки юзабилити мобильных интерфейсов: соответствие платформенным гайдлайнам, скорость отклика, предсказуемость навигации, доступность; методы юзабилити-тестирования: когнитивный проход, наблюдение за пользователем, A/B-тестирование макетов; стандарты WCAG и их применимость к мобильным приложениям, а также инструменты анализа: DevTools Flutter, accessibility inspectors в эмуляторах.		
			ПКос-5.2		Студент должен уметь: разрабатывать пользовательские сценарии и тест-кейсы для юзабилити-исследования; проводить экспертную оценку своего или чужого интерфейса по чек-листу: соот-	

					ветствие гайдлайнам, наличие обратной связи, обработка ошибок; использовать инструменты Flutter DevTools для анализа производительности и доступности, а также фиксировать замечания в виде отчёта с рекомендациями по улучшению.	
			ПКос-5.3			Студент должен владеть: навыками анализа полноты покрытия пользовательских сценариев в интерфейсе, определять, какие действия пользователя остаются протестированными; методикой выбора регистрируемых параметров при юзабилити-тестировании: время выполнения задачи, количество ошибок, субъективная оценка сложности; практиками итеративного улучшения интерфейса на основе обратной связи от реальных тестировщиков.
3.	ПКос-7	Способен осуществлять концептуально-логическое проектирование системы, разрабатывать техническое задание	ПКос-7.1	Студент должен знать: особенности предметной области АПК, требующие мобильных решений: учёт техники, мониторинг полей, регистрация урожая и соответствующие пользовательские роли; основы интеграции мо-		

				<p>бильного клиента с внешними сервисами: RESTful API, форматы данных (JSON), механизмы аутентификации (токены). платформенные ограничения, особенности работы с фоном в iOS/Android, политики безопасности, требования магазинов приложений к доступу к данным устройства.</p>		
			ПКос-7.2		<p>Студент должен уметь:          разрабатывать тест-планы для интеграционного тестирования: проверка корректности запросов к backend, обработки ошибок сети, сохранения данных при отсутствии связи;          готовить тестовые данные: mock-серверы, фейковые токены;          интерпретировать бизнес-требования и переводить их в тестовые сценарии;          работать в команде: согласовывать формат API с backend-разработчиками, сообщать об ошибках через issue-трекеры.</p>	
			ПКос-7.3			<p>Студент должен владеть:          навыками разработки стратегии интеграционного тестирования:</p>

						<p>определение критических точек взаимодействия, выбор инструментов, приоритизация сценариев;</p> <p>практиками управления процессом тестирования: ведение чек-листов, отслеживание дефектов, верификация исправлений;</p> <p>навыками документирования результатов тестирования в формате, понятном как разработчикам, так и заказчику.</p>
--	--	--	--	--	--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 4. Структура и содержание дисциплины

### 4.1 Распределение трудоёмкости дисциплины по видам работ по семестрам

Общая трудоёмкость дисциплины составляет 4 зач.ед. (144 часов), их распределение по видам работ и семестрам представлено в таблице 2.

Таблица 2

#### Распределение трудоёмкости дисциплины по видам работ

Вид учебной работы	Трудоёмкость
	час. всего
Общая трудоёмкость дисциплины по учебному плану	144
<b>1. Контактная работа:</b>	<b>53,4</b>
<b>Аудиторная работа</b>	<b>53,4</b>
лекции (Л)	16
практические занятия (ПЗ)	32
курсовой проект (КП) (консультация, защита)	3
консультации перед экзаменом	2
контактная работа на промежуточном контроле (КРА)	0,4
<b>2. Самостоятельная работа (СРС)</b>	<b>63,6</b>
самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к практическим занятиям)	63,6
Подготовка к экзамену (контроль)	27
Вид промежуточного контроля:	экзамен

### 4.2 Содержание дисциплины

Таблица 3

#### Тематический план учебной дисциплины

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ всего	ПКР	
Тема 1. «Введение в мобильную разработку и экосистему Flutter»	4,6	1	-		3,6
Тема 2 «Основы Dart и архитектура Flutter»	10	2	2		6
Тема 3 «Виджеты и компоновка»	16	2	6		8
Тема 4 «Навигация и маршрутизация»	11	1	4		6
Тема 5 «Управление состоянием»	23	4	7		12
Тема 6 «Работа с backend и локальным хранилищем»	17	2	5		10
Тема 7 «Доступ к возможностям устройства»	18	2	6		10
Тема 8 «Тестирование, сборка и публикация»	12	2	2		8
Курсовой проект (КП) (консультация, защита)	3	-	-	3	-

Наименование разделов и тем дисциплин (укрупнённо)	Всего	Аудиторная работа			Внеаудиторная работа СР
		Л	ПЗ всего	ПКР	
Подготовка к экзамену (контроль)	27	-	-	-	27
контактная работа на промежуточном контроле (КРА)	0,4	-	-	0,4	-
Консультации перед экзаменом	2	-	-	2	-
<b>Итого по дисциплине</b>	<b>144</b>	<b>16</b>	<b>32</b>	<b>5,4</b>	<b>90,6</b>

### **Введение в мобильную разработку и экосистему Flutter.**

Сравнение подходов к мобильной разработке: нативная (Kotlin/Swift), кроссплатформенная (Flutter, React Native, .NET MAUI) – их сильные стороны, ограничения и типичные сценарии применения. Обзор архитектуры Flutter: реактивный UI, компиляция в нативный код, hot reload. Установка и настройка SDK, структура проекта, управление зависимостями через pubspec.yaml, выбор IDE (Android Studio, VS Code).

### **Основы Dart и архитектура Flutter.**

Синтаксис языка Dart: типизация, null safety, асинхронность (async/await, Future, Stream), классы и наследование. Архитектурные принципы Flutter: виджет как описание UI, дерево виджетов и элементов, жизненный цикл StatefulWidget, контекст и зависимость от родителя. Принцип «всё есть виджет», иммутабельность, реактивное обновление интерфейса.

### **Виджеты и компоновка.**

Базовые и составные виджеты: Text, Button, Image, Icon, TextField. Системы компоновки: Row, Column, Stack, Expanded, SizedBox, ListView, GridView. Темизация: Material и Cupertino, кастомные темы, цвета, шрифты. Создание повторно используемых кастомных виджетов, анимации (AnimatedContainer, Hero), работа с ограничениями (constraints).

### **Навигация и маршрутизация.**

Базовая навигация: Navigator.push() / pop(), MaterialPageRoute. Передача данных между экранами, возврат результата. Именованные роуты, onGenerateRoute. Использование BottomNavigationBar, Drawer, Tabs. Обзор продвинутой навигации: пакет go\_router, глубокие ссылки (deep linking), состояние навигации.

### **Управление состоянием.**

Ограничения setState() в сложных приложениях. Паттерн Provider: ChangeNotifier, Consumer, context.watch/read/listen. Архитектура Bloc: события, состояния, bloc-классы, Cubit. Сравнение подходов: когда использовать setState, Provider или Bloc. Инъекция зависимостей, обработка ошибок и загрузочных состояний, реактивное обновление данных.

### **Работа с backend и локальным хранилищем.**

HTTP-запросы с использованием пакетов http и dio: GET, POST, обработка ошибок, таймауты. Аутентификация: передача токенов в заголовках, сохранение сессии. Локальное хранение: shared\_preferences для простых данных, Hive для объектов. Кэширование ответов, работа с JSON (вручную и через code generation), сериализация/десериализация.

### Доступ к возможностям устройства.

Использование пакетов из pub.dev для доступа к функциям устройства: камера (camera), геолокация (geolocator), уведомления (flutter\_local\_notifications), файлы, контакты, датчики. Обработка разрешений на Android и iOS. Эмуляция функций в симуляторе. Интеграция с картами через WebView или пакеты.

### Тестирование, сборка и публикация.

Виды тестов в Flutter: unit-тесты (логика), widget-тесты (UI), интеграционные тесты (сквозные сценарии). Написание тестов с использованием flutter\_test, мокирование зависимостей. Сборка приложения: flutter build apk / appbundle / ipa. Подготовка ресурсов: иконка, splash screen, локализация. Обзор требований для публикации.

## 4.3 Практические занятия

Таблица 4

### Содержание практических занятий и контрольные мероприятия

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
Тема 1. «Введение в мобильную разработку и экосистему Flutter»	Лекция 1. «Сравнение стеков мобильной разработки и основы Flutter»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
Тема 2. «Основы Dart и архитектура Flutter»	Лекция 2. «Синтаксис Dart: типы, асинхронность, null safety»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Лекция 3. «Архитектура Flutter: виджеты, дерево элементов, жизненный цикл»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 1. «Написание Dart-скриптов и первое Flutter-приложение»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	2
Тема 3. «Виджеты и компоновки»	Лекция 4. «Базовые и компоновочные виджеты»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3,	-	1

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ПКос-7.1, ПКос-7.2, ПКос-7.3.		
	Лекция 5. «Кастомные виджеты, темы и анимации»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 2. «Создание многоуровневого UI с компоновкой»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос -	3
	Практическая работа 3. «Разработка кастомного темизированного интерфейса»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос -	3
Тема 4. «Навигация и маршрутизация»	Лекция 6. «Навигация, передача данных и именованные роуты»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 4. «Реализация многоэкранного приложения с навигацией»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	2
	Практическая работа 5. «Добавление BottomNavigationBar и Drawer»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	2
Тема 5. «Управление состоянием»	Лекция 7. «Ограничения setState и введение в управление состоянием»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Лекция 8. «Паттерн Provider: ChangeNotifier и реактивность»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3,	-	1

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ПКос-7.1, ПКос-7.2, ПКос-7.3.		
	Лекция 9. «Архитектура Bloc: события, состояния, Cubit»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Лекция 10. «Сравнение подходов и выбор архитектуры под задачу»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 6. «Реализация приложения с setState и Provider»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос -	3
	Практическая работа 7. «Переход на Bloc: рефакторинг и расширение функционала»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	4
Тема 6. «Работа с backend и локальным хранилищем»	Лекция 11. «HTTP-запросы, аутентификация и работа с JSON»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Лекция 12. «Локальное хранение: shared_preferences и Hive»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 8. «Интеграция с fake REST API и кэширование»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	3
	Практическая работа 9. «Сохранение данных в Hive и восстановление при старте»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3,	Устный опрос	2

Название раздела, темы	№ и название лекций/ практических занятий	Формируемые компетенции (индикаторы)	Вид контрольного мероприятия	Кол-во часов
		ПКос-7.1, ПКос-7.2, ПКос-7.3.		
Тема 7. «Доступ к возможностям устройства»	Лекция 13. «Работа с камерой и файлами»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Лекция 14. «Геолокация, уведомления и разрешения»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 10. «Захват фото и сохранение в галерею»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	3
	Практическая работа 11. «Отображение координат и отправка уведомлений»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	3
Тема 8. «Тестирование, сборка и публикация»	Лекция 15. «Виды тестов в Flutter и написание widget-тестов»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Лекция 16. «Сборка приложения и подготовка к публикации»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	-	1
	Практическая работа 12. «Написание тестов и сборка debug-версии APK»	ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.	Устный опрос	2

**Перечень вопросов для самостоятельного изучения дисциплины**

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
1.	Тема 1. «Введение в мобильную разработку и экосистему Flutter»	Сравнение Flutter, React Native и .NET MAUI: преимущества и ограничения каждого стека для enterprise- и consumer-приложений. Принцип работы Dart VM и АОТ-компиляции в Flutter. Роль файла pubspec.yaml в управлении зависимостями и ресурсами. Установка и настройка Flutter SDK на Windows, macOS и Linux. Основные команды CLI: flutter doctor, flutter create, flutter run. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).
2.	Тема 2. «Основы Dart и архитектура Flutter»	Разница между Future и Stream в Dart и их применение в мобильной разработке. Как null safety влияет на безопасность кода и обработку опциональных значений. Почему виджет в Flutter – это не объект, а функция описания UI. Особенности жизненного цикла StatefulWidget: initState, dispose, didUpdateWidget. Роль BuildContext и почему его нельзя сохранять в переменную. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).
3.	Тема 3. «Виджеты и компоновки»	Почему в Flutter нельзя задать абсолютные координаты виджета (в отличие от CSS). Как работают ограничения (constraints) и почему возникает ошибка «RenderBox was not laid out». Разница между Expanded и Flexible в Row/Column. Как создать кастомный виджет, переиспользуемый в разных частях приложения. Особенности темизации: как переопределить цвет кнопки глобально через ThemeData. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).
4.	Тема 4. «Навигация и маршрутизация»	Когда использовать именованные роуты, а когда – анонимную навигацию. Как передать данные в новый экран и получить результат при возврате. Почему Navigator.pushReplacement используется при переходе с экрана авторизации. Ограничения навигации в Flutter без использования сторонних пакетов. Как реализовать глубокие ссылки (deep linking) в приложении. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).
5.	Тема 5. «Управление состоянием»	Почему setState не подходит для крупных приложений и как это приводит к «ненужным» перестроениям. Как Provider использует InheritedWidget под капотом для эффективной передачи данных. В чём разница между Bloc и Cubit и когда что использовать. Как организовать обработку ошибок и загрузочных состояний в паттерне Bloc. Как избежать утечки памяти при использовании потоков (Stream) в состояниях. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).

№ п/п	Название раздела, темы	Перечень рассматриваемых вопросов для самостоятельного изучения
6.	Тема 6. «Работа с backend и локальным хранилищем»	Как правильно обрабатывать сетевые ошибки (таймаут, отсутствие интернета) в Dio. Зачем использовать json_serializable вместо ручного парсинга JSON. Как безопасно хранить токены аутентификации на устройстве. Почему shared_preferences не подходит для хранения чувствительных данных или больших объектов. Как организовать кэширование API-ответов без дублирования данных. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).
7.	Тема 7. «Доступ к возможностям устройства»	Как запросить разрешения на геолокацию в Android 10+ и iOS 14+. Почему камера может не работать в эмуляторе и как это обойти. Как отправить локальное уведомление без backend-сервера. Как получить путь к папке «Документы» на разных платформах. Какие ограничения накладывает Apple на использование датчиков и камеры в фоне. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).
8.	Тема 8. «Тестирование, сборка и публикация»	В чём разница между widget-тестом и интеграционным тестом в Flutter. Как мокировать HTTP-клиент при написании unit-тестов. Почему flutter build apk создаёт debug-версию, а --release – production. Как добавить иконку и splash screen без ручного редактирования native-файлов. Какие минимальные метаданные требуются для публикации. (ПКос-3.1, ПКос-3.2, ПКос-3.3, ПКос-5.1, ПКос-5.2, ПКос-5.3, ПКос-7.1, ПКос-7.2, ПКос-7.3.).

## 5. Образовательные технологии

Таблица 6

### Применение активных и интерактивных образовательных технологий

№ п/п	Тема и форма занятия	Наименование используемых активных и интерактивных образовательных технологий (форм обучения)
1.	Тема 2. «Основы Dart и архитектура Flutter»	ПЗ Мозговой штурм
2.	Тема 4. «Навигация и маршрутизация»	ПЗ Разбор конкретных ситуаций
3.	Тема 7. «Доступ к возможностям устройства»	ПЗ Разбор конкретных ситуаций
4.	Тема 8. «Тестирование, сборка и публикация»	ПЗ Разбор конкретных ситуаций

## **6. Текущий контроль успеваемости и промежуточная аттестация по итогам освоения дисциплины**

### **6.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений и навыков и (или) опыта деятельности**

#### **Вопросы для подготовки к устным опросам**

##### **Тема 1. Введение в мобильную разработку и экосистему Flutter**

1. Какие три основных подхода к мобильной разработке вы знаете?
2. Чем Flutter отличается от React Native по способу отрисовки UI?
3. Какая команда в терминале проверяет, всё ли установлено для работы с Flutter?
4. Что указывается в файле pubspec.yaml?
5. Какой IDE вы используете для разработки и почему?

##### **Тема 2. Основы Dart и архитектура Flutter**

1. Как объявить переменную, которая может быть null в Dart с включённым null safety?
2. Что делает ключевое слово async перед функцией?
3. Чем StatelessWidget отличается от StatefulWidget?
4. Какой метод вызывается при создании состояния виджета?
5. Что такое BuildContext и где он используется?

##### **Тема 3. Виджеты и компоновка**

1. Какой виджет вы используете, чтобы расположить элементы по вертикали?
2. Как сделать так, чтобы кнопка заняла всё свободное пространство в строке?
3. Как изменить цвет фона всего приложения?
4. Как создать кастомный виджет, который можно использовать несколько раз?
5. Какой виджет используется для отображения списка прокручиваемых элементов?

##### **Тема 4. Навигация и маршрутизация**

1. Как перейти на новый экран в Flutter?
2. Как вернуться на предыдущий экран?
3. Как передать данные в новый экран?
4. Как использовать BottomNavigationBar для переключения между экранами?
5. Что такое именованные роуты и зачем они нужны?

##### **Тема 5. Управление состоянием**

1. Что делает метод setState()?
2. Какой пакет вы использовали для управления состоянием через Provider?
3. Что такое ChangeNotifier и как он связан с Provider?
4. Как в Bloc определяется новое состояние после события?
5. Почему setState не подходит для сложных приложений?

##### **Тема 6. Работа с backend и локальным хранилищем**

1. Какой пакет вы использовали для отправки HTTP-запросов?

2. Как сохранить простое значение (например, строку) на устройстве?
3. Где хранятся токены аутентификации в вашем приложении?
4. Как обработать ошибку, если сервер не отвечает?
5. Что такое Hive и зачем он нужен?

#### **Тема 7. Доступ к возможностям устройства**

1. Какой пакет вы использовали для работы с камерой?
2. Как получить текущие координаты устройства?
3. Как запросить разрешение на геолокацию?
4. Как отправить локальное уведомление без интернета?
5. Где сохраняются фото, сделанные через приложение?

#### **Тема 8. Тестирование, сборка и публикация**

1. Какой вид теста вы писали для проверки UI-элемента?
2. Какая команда собирает APK-файл в debug-режиме?
3. Как добавить иконку в приложение?
4. Где описывается название приложения для магазина?
5. Что нужно сделать, чтобы тест «нашёл» кнопку по тексту?

**Критерии оценки:** опрос осуществляет из всего списка студентов. За каждый вопрос, студент может получить максимум 10 баллов. 1 студент, в течение устного опроса отвечает только на 1 вопрос. Таким образом, за каждый опрос студент может получить максимум 10 баллов.

#### **Перечень вопросов, выносимых на промежуточную аттестацию (экзамен)**

1. Почему Flutter использует собственный движок рендеринга, а не нативные компоненты? Как это влияет на производительность и внешний вид?
2. В каких случаях предпочтительнее выбрать .NET MAUI вместо Flutter при разработке для enterprise-сектора?
3. Почему виджеты в Flutter иммутабельны? Как это связано с производительностью?
4. Что произойдёт, если вызвать setState() после dispose()? Как избежать этой ошибки?
5. Как Dart-код преобразуется в нативный на iOS и Android? В чём разница между JIT и AOT?
6. Почему использование глобальных переменных в Flutter считается плохой практикой?
7. Почему возникает ошибка «BoxConstraints forces an infinite height» и как её исправить?
8. Как правильно вложить ListView внутрь Column, чтобы избежать ошибок компоновки?
9. В чём преимущество кастомных композитных виджетов перед функциями-хелперами?
10. Как реализовать адаптивный интерфейс, который корректно отображается и на мобильном, и на планшете?
11. Почему при использовании Navigator.push() без сохранения ссылки нельзя программно закрыть экран?

12. Как реализовать «глубокую» навигацию (например, от push-уведомления сразу в детали записи)?
13. В чём недостаток анонимной навигации при построении сложных маршрутов?
14. Как синхронизировать состояние навигации с бизнес-логикой (например, заблокировать переход при незавершённой загрузке)?
15. Почему при использовании Provider с ChangeNotifier важно вызывать notifyListeners() только после изменения данных?
16. Как избежать «лишних» перестроений виджетов при использовании context.watch()?
17. В чём разница между Bloc и Cubit с точки зрения тестируемости и читаемости кода?
18. Как организовать обработку асинхронного события (например, загрузка данных) в Bloc без блокировки UI?
19. Почему setState может привести к утечке памяти при работе с потоками или таймерами?
20. Почему хранение JWT-токена в shared\_preferences небезопасно и какие есть альтернативы?
21. Как организовать retry-механизм при неудачном HTTP-запросе с экспоненциальной задержкой?
22. В чём преимущество json\_serializable перед ручным парсингом JSON с точки зрения поддержки кода?
23. Как избежать дублирования данных при одновременном использовании кэширования и локального хранилища?
24. Как реализовать «оффлайн-first» архитектуру в мобильном приложении?
25. Почему на iOS требуется указывать NSLocationWhenInUseUsageDescription в Info.plist? Что произойдёт, если этого не сделать?
26. Как обработать ситуацию, когда пользователь отклонил запрос на доступ к камере — и как повторно запросить разрешение?
27. Как отправить уведомление через flutter\_local\_notifications, если приложение закрыто?
28. Почему работа с файлами на Android 10+ требует использования MediaStore вместо прямого пути?
29. Как получить список доступных датчиков на устройстве и подписаться на их показания?
30. В чём разница между pump() и pumpAndSettle() в widget-тестах?
31. Как протестировать поведение виджета при изменении состояния, управляемого через Provider?
32. Почему flutter build apk без флага --split-per-abi создаёт слишком большой APK?
33. Как включить отладку в release-сборке для диагностики ошибок у пользователей?
34. Как выбрать между Provider и Bloc для приложения с 5 экранами и простой логикой?

35. Почему при работе с геолокацией и картами часто используют WebView вместо нативных SDK?
36. Как организовать архитектуру, чтобы легко заменить REST API на GraphQL в будущем?
37. Какие метрики производительности (fps, memory) стоит отслеживать при доработке UI?
38. Как обеспечить, что ваше приложение корректно освобождает ресурсы (камера, потоки) при закрытии?

### **Практическое задание № 1**

#### **«Написание Dart-скриптов и первое Flutter-приложение»**

Цель: изучить основы языка Dart и архитектуры Flutter, освоить создание простого приложения с использованием базовых виджетов и асинхронных операций.

Условие: требуется разработать консольный Dart-скрипт для обработки данных и мобильное приложение для отображения приветствия с текущей датой.

Требуется, используя Dart и Flutter:

1. Создать Dart-скрипт (вне Flutter):
  - a. Создать файл `utils.dart` и реализовать функцию `formatDate(DateTime date)`, возвращающую строку вида «10 января 2026».
  - b. Реализовать функцию `isWeekend(DateTime date)`, возвращающую `true`, если дата – суббота или воскресенье.
  - c. Написать `main()`, который выводит текущую дату и указывает, является ли она выходным днём.
2. Создать первое Flutter-приложение:
  - a. Инициализировать проект через `flutter create first_app`.
  - b. Заменить содержимое `main.dart` на `MaterialApp` с `Scaffold`.
  - c. В центре экрана отобразить текст: «Добро пожаловать! Сегодня: [дата]», используя функцию `formatDate` из п. 1a.
  - d. Под текстом добавить кнопку «Проверить выходной», при нажатии на которую появляется `SnackBar` с сообщением «Сегодня выходной!» или «Рабочий день».
3. Протестировать работу:
  - a. Запустить Dart-скрипт в терминале.
  - b. Запустить Flutter-приложение в эмуляторе или на устройстве.
  - c. Убедиться, что дата отображается корректно и кнопка работает.

### **Практическое задание № 2**

#### **«Создание многоуровневого UI с компоновкой»**

Цель: освоить принципы компоновки виджетов в Flutter, научиться создавать адаптивные и структурированные интерфейсы с использованием `Row`, `Column`, `ListView` и других `layout`-виджетов.

Условие: требуется разработать экран «Карточка техники» для приложения учёта сельхозтехники.

Требуется:

1. Создать структуру экрана:
  - a. Верхняя часть – изображение техники (использовать `Image.asset` или `Image.network` с `placeholder`).
  - b. Средняя часть – форма с полями:
    - Название (текст),
    - Модель (выпадающий список: Трактор, Комбайн, Сеялка),
    - Инвентарный номер (поле ввода),
    - Дата последнего ТО (текст, например, «10.01.2026»).
  - c. Нижняя часть – две кнопки: «Редактировать» и «Удалить», расположенные горизонтально.
2. Реализовать компоновку:
  - a. Использовать `Column` как основной контейнер.
  - b. Поля формы организовать через `ListView` или вложенные `Row` с `Expanded`.
  - c. Кнопки разместить в `Row` с `MainAxisAlignment.spaceEvenly`.
3. Добавить базовую интерактивность:
  - a. При нажатии на «Редактировать» – показывать `SnackBar`: «Режим редактирования включён».
  - b. При нажатии на «Удалить» – показывать `AlertDialog` с подтверждением удаления.
4. Протестировать:
  - a. Проверить, что интерфейс не выходит за границы экрана на разных разрешениях.
  - b. Убедиться, что все элементы корректно выравниваются и реагируют на нажатия.

## **Комплект разноуровневых заданий**

### **Тема 1 «Введение в мобильную разработку и экосистему Flutter»**

#### **Задание репродуктивного уровня**

Перечислите три основных подхода к мобильной разработке и укажите, какой из них использует Flutter. В чём состоит главный принцип работы Flutter-движка?

#### **Задание реконструктивного уровня**

Студент установил Flutter, но команда `flutter doctor` показывает ошибки для Android Studio и лицензий. Опишите пошагово, как исправить типичные проблемы: отсутствие SDK, неправильный путь к лицензии, эмулятор не запускается.

### **Тема 2 «Основы Dart и архитектура Flutter»**

#### **Задание репродуктивного уровня**

Опишите разницу между `StatelessWidget` и `StatefulWidget`. Какой метод вызывается при создании состояния, и какой – при уничтожении?

#### **Задание реконструктивного уровня**

В приложении возникает ошибка: «`setState() called after dispose()`». Объясните, почему это происходит, и предложите два способа предотвратить эту ошибку при работе с асинхронными операциями.

### **Тема 3 «Виджеты и компоновка»**

Задание репродуктивного уровня

Перечислите виджеты, которые используются для вертикальной и горизонтальной компоновки в Flutter. Какой виджет позволяет прокручивать содержимое?

Задание реконструктивного уровня

При вложении ListView внутрь Column возникает ошибка «Vertical viewport was given unbounded height». Объясните причину и предложите два способа исправить компоновку.

### **Тема 4 «Навигация и маршрутизация»**

Задание репродуктивного уровня

Опишите, как передать данные в новый экран с помощью Navigator.push(). Как получить результат при возврате с этого экрана?

Задание реконструктивного уровня

В приложении после авторизации нужно перейти на главный экран и заблокировать возврат к форме входа. Опишите, как реализовать такое поведение с использованием Navigator.pushReplacement().

### **Тема 5 «Управление состоянием»**

Задание репродуктивного уровня

Опишите, как Provider передаёт данные от ChangeNotifier к виджету. Какой метод вызывается для уведомления об изменении состояния?

Задание реконструктивного уровня

В приложении на Provider при обновлении списка задач перестраиваются все виджеты, включая AppBar. Объясните, почему это происходит, и предложите способ оптимизировать перестроение, затронув только нужную часть UI.

### **Тема 6 «Работа с backend и локальным хранилищем»**

Задание репродуктивного уровня

Какие пакеты в экосистеме Flutter используются для HTTP-запросов и локального хранения объектов? Назовите по одному примеру каждого.

Задание реконструктивного уровня

Приложение получает данные с сервера, но при отсутствии интернета показывает пустой экран. Предложите архитектурное решение, позволяющее показывать кэшированные данные из Hive, если сеть недоступна.

### **Тема 7 «Доступ к возможностям устройства»**

Задание репродуктивного уровня

Какие пакеты из pub.dev используются для работы с камерой и геолокацией? Какие разрешения нужно запросить в Android и iOS для доступа к геолокации?

Задание реконструктивного уровня

Студент реализовал захват фото, но на iOS приложение падает. Объясните, какая настройка в Info.plist обязательна для работы с камерой, и как корректно обработать отказ пользователя в предоставлении разрешения.

### **Тема 8 «Тестирование, сборка и публикация»**

Задание репродуктивного уровня

Какие три типа тестов поддерживаются в Flutter? Для чего предназначен каждый из них?

Задание реконструктивного уровня

При сборке flutter build apk получается файл размером 150 МБ. Объясните, почему это происходит, и предложите два способа уменьшить размер релизной сборки.

**Критерии оценки:** принимаются преподавателем и оцениваются вместе с устными опросами.

## **6.2. Описание показателей и критериев контроля успеваемости, описание шкал оценивания**

Для оценки знаний, умений, навыков и формирования компетенции по дисциплине применяется **балльно-рейтинговая** система контроля и оценки успеваемости студентов.

В основу балльно-рейтинговой системы (БРС) положены принципы, в соответствии с которыми формирование рейтинга студента осуществляется в ходе текущей работы в семестре.

Работы должны быть выполнены по своему варианту, оформлены в соответствии с требованиями стандартов по оформлению текстовых документов в текстовом редакторе MS Word. Работы сдаются в электронном виде.

По результатам защиты могут быть получены следующие баллы:

9-10 баллов – расчеты (если имеются) проведены корректно, результаты правильно интерпретированы. Полностью выполнены все пункты выданного задания. Работа оформлена в соответствии с требованиями стандартов по оформлению текстовых документов. Студент развернуто и свободно ответил на все вопросы при защите работы.

7-8 баллов – работа выполнена, выполнены все пункты выданного задания, но не полностью, либо с несущественными ошибками, имеются незначительные ошибки в интерпретации результатов и/или оформлении. Студент в целом ответил на все поставленные вопросы, ориентируется в работе.

4-6 баллов – работа в целом выполнена, выполнены основные, но не все пункты выданного задания, либо с существенными ошибками, имеются значительные ошибки в интерпретации полученных результатов и представления данных, оформления работы. Некоторые вопросы по работе вызывают затруднения.

1-3 балла – имеются грубые ошибки в методике выполнения, интерпретации полученных результатов и представления данных, оформления работы, большая часть пунктов выданного задания не выполнена. Студент не отвечает на вопросы при защите.

В течение периода обучения по дисциплине студент должен выполнить и защитить 11 практических заданий (индивидуальных или групповых проектов), каждое из которых оценивается максимум на 10 баллов. За посещение занятий добавляется 0,15 балла за каждый час ( $68 \cdot 0,15$ ), участие в конференции с до-

кладом по теме, связанной с возможностями практического применения языка Python – 10 баллов. Таким образом, максимально возможная сумма баллов равна:  $11 \cdot 10 + 68 \cdot 0,15 + 10 = 110 + 10 + 10 = 130$ .

Зачет по дисциплине получают студенты, набравшие не менее 60% от максимального количества баллов, т.е. 78 баллов и более.

Итоговая оценка по дисциплине выставляется преподавателем в соответствии со шкалой:

Текущий рейтинг	Оценка			
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
в процентах	0-59	60-69	70-84	85-100
в баллах	0-77	78-90	91-110	111-130

Студенты, набравшие в течение семестра менее 78 баллов, пишут итоговую зачетную работу. К написанию итоговой зачетной работы допускаются студенты, **в случае выполнения всех практических работ**.

## 7. Учебно-методическое и информационное обеспечение дисциплины

### 7.1 Основная литература

1. Калгина, И. С. Разработка мобильных приложений : учебное пособие / И. С. Калгина. – Чита : ЗабГУ, 2022. – 163 с. – ISBN 978-5-9293-3137-4. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/363323> – Режим доступа: для авториз. пользователей.

2. Бунина, Л. В. Разработка мобильных приложений : учебное пособие / Л. В. Бунина, А. П. Титов. – Москва : РТУ МИРЭА, 2025 – Часть 1 – 2025. – 123 с. – ISBN 978-5-7339-2604-9. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/504863> – Режим доступа: для авториз. пользователей.

3. Чернов, Е. А. Проектирование и разработка мобильных приложений. Часть 1 : учебное пособие / Е. А. Чернов. – Москва : РТУ МИРЭА, 2025. – 69 с. – ISBN 978-5-7339-2652-0. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/507485> – Режим доступа: для авториз. пользователей.

### 7.2 Дополнительная литература

1. Заметти, Ф. Flutter на практике : руководство / Ф. Заметти ; перевод с английского А. С. Тищенко. – Москва : ДМК Пресс, 2020. – 328 с. – ISBN 978-5-97060-808-1. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/179464> – Режим доступа: для авториз. пользователей.

2. Климченко, К. П. Разработка пользовательских интерфейсов : методические рекомендации / К. П. Климченко, Р. А. Исаев, Р. С. Толмасов. – Москва : РТУ МИРЭА, 2022. – 25 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/310874> – Режим доступа: для авториз. пользователей.

3. Богданова, С. В. Технологии программирования : учебник / С. В. Богданова, А. Н. Ермакова. – Ставрополь : СтГАУ, 2024 – Часть 1 – 2024. – 236 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/510225> – Режим доступа: для авториз. пользователей.

4. Васильев, Н. П. Введение в гибридные технологии разработки мобильных приложений / Н. П. Васильев, А. М. Заяц. – 3-е изд., стер. – Санкт-Петербург : Лань, 2022. – 160 с. – ISBN 978-5-507-44502-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/230387> – Режим доступа: для авториз. пользователей.

5. Баланов, А. Н. Комплексное руководство по разработке: от мобильных приложений до веб-технологий : учебное пособие для вузов / А. Н. Баланов. – 2-е изд., стер. – Санкт-Петербург : Лань, 2025. – 412 с. – ISBN 978-5-507-53193-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/478178> – Режим доступа: для авториз. пользователей.

## **8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

1. Основы Dart. – URL: <https://stepik.org/course/92982/promo?search=8585688873>

2. Введение во Flutter. – URL: <https://stepik.org/course/118419/promo?search=8585688872>

3. Flutter online. – URL: <https://flutlab.io/>

4. Яндекс образование. Коротко о Dart: часть первая – URL: <https://education.yandex.ru/handbook/flutter/article/korotko-o-dart-chast-pervaia>

5. Цифровые профессии: Искусственный интеллект. – URL: <https://steps.2035.university/collections/f6361b9a-ea2e-41b1-a18f-9a2f84a9fcd4>

6. Kaggle. – URL: <https://www.kaggle.com/>

7. Dart programming language – URL: <https://dart.dev/>

8. VS code – URL: <https://code.visualstudio.com/>

9. Flutter – URL: <https://flutter.dev/>

## 9. Перечень программного обеспечения и информационных справочных систем

Таблица 9

### Перечень программного обеспечения

№ п/п	Наименование раздела учебной дисциплины	Наименование программы	Тип программы	Автор	Год разработки
1	Тема 1. «Введение в веб-разработку и архитектура современных приложений»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
2	Тема 2 «Разработка на Flask: от простого приложения до продвинутых паттернов»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
3	Тема 3 «Продвинутое приложение на Django»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
4	Тема 4 «Современные API с FastAPI и асинхронностью»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012
5	Тема 5 «Тестирование, безопасность и развертывание веб-приложений»	Excel/ Word/Anaconda	Расчетная/система управления пакетами и дистрибутив	Microsoft/Anaconda Inc.	2007/2012

## 10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Таблица 10

### Сведения об обеспеченности специализированными аудиториями, кабинетами, лабораториями

Наименование специальных* помещений и помещений для самостоятельной работы (№ учебного корпуса, № аудитории)	Оснащенность специальных помещений и помещений для самостоятельной работы**
1	2
учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивиду-	1. Компьютер – 29 шт.; 2. Стенд «Сергеев Сергей Степанович 1910-1999» (Инв.№591013/25) – 1 шт.; 3. Огнетушитель порошковый (Инв. №559527) – 1 шт.; 4. Подвесное крепление к огнетушителю (Инв. № 559528) – 1 шт.;

альных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации (2й учебный корпус, 102 ауд.)	<ul style="list-style-type: none"> <li>5. Жалюзи (Инв. №1107-221225, Инв. №1107-221225) – 2шт.;</li> <li>6. Стул – 29 шт.;</li> <li>7. Стол компьютерный – 28 шт.;</li> <li>8. Стол для преподавателя – 1 шт.;</li> <li>9. Доска маркерная (Инв. № 558762/5) – 1 шт.;</li> <li>10. Трибуна напольная (без инв. №) – 1 шт.</li> </ul>
учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 106 ауд.)	<ul style="list-style-type: none"> <li>1. Рабочая станция FORSITE TH1516G512G, Российская Федерация A4Tech Fstyler F1512 – 16 шт.;</li> <li>2. Стол наборный (Инв. №410136000010828) – 1 шт.</li> <li>3. Стол компьютерный (Инв. № 410136000010813-410136000010827) – 15 шт.;</li> <li>4. Стул (Инв. № 410136000010829-410136000010853) – 25 шт.;</li> <li>5. Интерактивная панель (Инв. № 410124000603715) – 1 шт.</li> </ul>
учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 302 ауд.)	<ul style="list-style-type: none"> <li>1. Компьютер – 16 шт.</li> <li>2. Телевизор – 1 шт.</li> <li>3. Стол для преподавателя – 1 шт.</li> <li>4. Стол компьютерный – 16 шт.</li> <li>5. Стул офисный – 17 шт.</li> <li>6. Компьютер: PRO-3159209 Intel Core i5-10400 2900МГц, Intel B460, 16Гб DDR4, Intel UHD Graphics 630 (встроенная), SSD 240Гб, 500Вт, Mini-Tower – 1 шт.</li> <li>7. Кондиционер HAIER HSU -24HPL03/R3 (Инв. № 210134000062198) – 1 шт.</li> <li>8. Вешалка напольная (Инв.№1107-333144, Инв.№1107-333144) – 2 шт.</li> </ul>
учебная аудитория для проведения занятий лекционного типа, учебная аудитория для проведения занятий семинарского типа, учебная аудитория для проведения курсового проектирования (выполнения курсовых работ), учебная аудитория для групповых и индивидуальных консультаций, учебная аудитория для текущего контроля и промежуточной аттестации, помещение для самостоятельной работы (2й учебный корпус, 303 ауд.)	<ul style="list-style-type: none"> <li>1. Трибуна напольная (Инв.№ 599206) – 1 шт.;</li> <li>2. Жалюзи (Инв.№591110) – 1 шт.;</li> <li>3. Доска маркетинговая (Инв.№ 35643/4) – 1 шт.;</li> <li>4. Стол – 15 шт.;</li> <li>5. Скамейка – 14 шт.;</li> <li>6. Стол эрго – 1 шт.;</li> <li>7. Стул – 16 шт.</li> </ul>
Центральная научная библиотека имени Н.И. Железнова	Читальные залы библиотеки
Студенческое общежитие	Комната для самоподготовки

## **11. Методические рекомендации студентам по освоению дисциплины**

Приступая к изучению дисциплины «Разработка мобильных приложений», студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке РГАУ-МСХА им. К.А. Тимирязева, получить в библиотеке рекомендованные учебники и учебно-методические пособия, завести новую тетрадь для работы с первоисточниками.

В ходе занятий вести конспектирование учебного материала. Обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к практическим занятиям изучить основную литературу, ознакомиться с дополнительной литературой в соответствии с поставленной задачей. При этом учесть рекомендации преподавателя и требования учебной программы. Необходимо дорабатывать свой конспект, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной учебной программой.

При подготовке к экзамену (в конце семестра) повторять пройденный материал в строгом соответствии с учебной программой. Использовать конспекты и литературу, рекомендованную преподавателем. Обратит особое внимание на темы учебных занятий, пропущенных студентом по разным причинам. При необходимости обратиться за консультацией и методической помощью к преподавателю.

### **Виды и формы отработки пропущенных занятий**

Студент, пропустивший занятия обязан самостоятельно подготовиться к теме устного опроса, которые состоялись на практическом занятии. В рамках часов консультаций студент может ответить на вопросы пропущенного устного опроса, которые были пропущены.

## **12. Методические рекомендации преподавателям по организации обучения по дисциплине**

Курс «Разработка мобильных приложений» должен давать не абстрактно-формальные, а прикладные знания. Данная цель может быть реализована только при условии соблюдения в учебных планах преемственности учебных дисциплин. Базовые знания для изучения Разработка мобильных приложений дают такие дисциплины, как Алгоритмизация и программирование, Разработка профессиональных приложений. Освоение основных тем данной дисциплины позволит студентам сформировать представление о таком сложном предмете как тестирование программного обеспечения, понять всю ширину науки и получить необходимые знания для последующего профессионального развития в этой области.

Студент может подготовить доклад по теме, представляющей его научный интерес, представить результаты в виде презентации. В случае надлежащего качества, его работа может быть заслушана на научном кружке кафедры или

на студенческой научной конференции. По решению кафедры, студенты, занявшие призовые места на научных студенческих конференциях, могут освободиться от сдачи экзамена по этой дисциплине.

Преподаватель должен указывать, в какой последовательности следует изучать материал дисциплины, обращать внимание на особенности изучения отдельных тем и разделов, помогать отбирать наиболее важные и необходимые сведения из учебных пособий, а также давать объяснения вопросам программы курса, которые обычно вызывают затруднения. При этом преподавателю необходимо учитывать следующие моменты:

1. Не следует перегружать студентов творческими заданиями.
2. Чередовать творческую работу на занятиях с заданиями во внеаудиторное время.
3. Давать студентам четкий инструктаж по выполнению самостоятельных заданий: цель задания; условия выполнения; объем; сроки; требования к оформлению.
4. Осуществлять текущий учет и контроль за самостоятельной работой.
5. Давать оценку и обобщать уровень усвоения навыков самостоятельной, творческой работы.

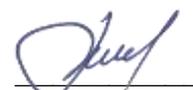
**Программу разработал:**

Демичев В.В., канд. экон. наук, доцент

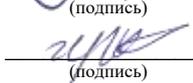
(ФИО, ученая степень, ученое звание)

Храмов Д.Э., ассистент

(ФИО, ученая степень, ученое звание)



(подпись)



(подпись)

## РЕЦЕНЗИЯ

### на рабочую программу дисциплины Б1.В.17 «Разработка мобильных приложений» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направ- ленность «Фуллстек разработка» (квалификация выпускника – бакалавр)

Вахрушевой Инны Алексеевны, доцентом кафедры высшей математики, кандидатом педагогических наук (далее по тексту рецензент), проведено рецензирование рабочей программы дисциплины «Разработка мобильных приложений» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Фуллстек разработка» (бакалавриат) разработанной в ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», на кафедре статистики и кибернетики (разработчики – Демичев Вадим Владимирович, доцент, кандидат экономических наук, Храмов Дмитрий Эдуардович, ассистент кафедры статистики и кибернетики).

Рассмотрев представленные на рецензирование материалы, рецензент пришел к следующим выводам:

1. Предъявленная рабочая программа дисциплины «Разработка мобильных приложений» (далее по тексту Программа) соответствует требованиям ФГОС ВО по направлению 09.03.02 Информационные системы и технологии. Программа содержит все основные разделы, соответствует требованиям к нормативно-методическим документам.

2. Представленная в Программе **актуальность** учебной дисциплины в рамках реализации ОПОП ВО не подлежит сомнению – дисциплина относится к дисциплинам части, формируемой участниками образовательных отношений – Б1.В

3. Представленные в Программе **цели** дисциплины соответствуют требованиям ФГОС ВО направления 09.03.02 Информационные системы и технологии.

4. В соответствии с Программой за дисциплиной «Разработка мобильных приложений» закреплены **3 профессиональные компетенции (ПКос)**. Дисциплина «Разработка мобильных приложений» и представленная Программа способна реализовать их в объявленных требованиях.

5. **Результаты обучения**, представленные в Программе в категориях знать, уметь, владеть соответствуют специфике и содержанию дисциплины и демонстрируют возможность получения заявленных результатов.

6. Общая трудоёмкость дисциплины «Разработка мобильных приложений» составляет 4 зачётные единицы (144 часа).

7. Информация о взаимосвязи изучаемых дисциплин и вопросам исключения дублирования в содержании дисциплин соответствует действительности. Дисциплина «Разработка мобильных приложений» взаимосвязана с другими дисциплинами ОПОП ВО и Учебного плана по направлению 09.03.02 Информационные системы и технологии и возможность дублирования в содержании отсутствует.

8. Представленная Программа предполагает использование современных образовательных технологий, используемые при реализации различных видов учебной работы. Формы образовательных технологий соответствуют специфике дисциплины.

9. Программа дисциплины «Разработка мобильных приложений» предполагает проведение занятий в интерактивной форме.

10. Виды, содержание и трудоёмкость самостоятельной работы студентов, представленные в Программе, соответствуют требованиям к подготовке выпускников, содержащимся во ФГОС ВО направления 09.03.02 Информационные системы и технологии.

11. Представленные и описанные в Программе формы *текущей* оценки знаний (устный опрос), соответствуют специфике дисциплины и требованиям к выпускникам.

Форма промежуточного контроля знаний студентов, предусмотренная Программой, осуществляется в форме экзамена в восьмом семестре, что соответствует статусу дисциплины, как дисциплины части, формируемой участниками образовательных отношений дисциплин учебного плана – Б1.В ФГОС ВО направления 09.03.02. Информационные системы и технологии.

12. Формы оценки знаний, представленные в Программе, соответствуют специфике дисциплины и требованиям к выпускникам.

13. Учебно-методическое обеспечение дисциплины представлено: основной литературой – 3 источника (базовые учебники), дополнительной литературой – 5 наименования, Интернет-ресурсы – 9 источника и соответствует требованиям ФГОС ВО направления 09.03.02 Информационные системы и технологии.

14. Материально-техническое обеспечение дисциплины соответствует специфике дисциплины «Разработка мобильных приложений» и обеспечивает использование современных образовательных, в том числе интерактивных методов обучения.

15. Методические рекомендации студентам и методические рекомендации преподавателям по организации обучения по дисциплине дают представление о специфике обучения по дисциплине «Разработка мобильных приложений».

#### **ОБЩИЕ ВЫВОДЫ**

На основании проведенного рецензирования можно сделать заключение, что характер, структура и содержание рабочей программы дисциплины «Разработка мобильных приложений» ОПОП ВО по направлению 09.03.02 Информационные системы и технологии, направленность «Фуллстек разработка» (квалификация выпускника – бакалавр), разработанная Демичевым Вадимом Владимировичем, доцентом, кандидатом экономических наук, Храмовым Дмитрием Эдуардовичем, ассистентом кафедры статистики и кибернетики соответствует требованиям ФГОС ВО, современным требованиям экономики, рынка труда и позволит при её реализации успешно обеспечить формирование заявленных компетенций.

Рецензент: Вахрушева Инна Алексеевна, доцент кафедры высшей математики ФГБОУ ВО «Российский государственный аграрный университет – МСХА имени К.А. Тимирязева», кандидат педагогических наук

  
(подпись)

«26» августа 2025 г.