

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»



УТВЕРЖДАЮ

и.о. ректора

М.В. Румянцев М.В. Румянцев

« 07 » 08 2025 г.

**Дополнительная профессиональная программа
(программа профессиональной переподготовки)**

«Мобильные приложения для электронной торговли: основы фреймворка
Flutter»

Красноярск 2025

I. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

I.1. Нормативная правовая основа Программы

Дополнительная профессиональная программа (программа профессиональной переподготовки) ИТ-профиля «Мобильные приложения для электронной торговли: основы фреймворка Flutter» (далее — Программа) разработана в соответствии с нормами Федерального закона РФ от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»; с постановлением Правительства РФ от 11 октября 2023 г. № 1678 «Об утверждении Правил применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»; приказом Минобрнауки России от 19 октября 2020 г. № 1316 «Об утверждении порядка разработки дополнительных профессиональных программ, содержащих сведения, составляющие государственную тайну, и дополнительных профессиональных программ в области информационной безопасности» (далее – приказ Минобрнауки России № 1316); методические рекомендации по разработке основных профессиональных образовательных программ и дополнительных профессиональных программ с учетом соответствующих профессиональных стандартов (утв. Минобрнауки России 22 января 2015 г. № ДЛ-1/05вн); с учетом требований приказа Минобрнауки России от 1 июля 2013 г. № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам», с изменениями, внесенными приказом Минобрнауки России от 15 ноября 2013 г. № 1244 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам, утвержденный приказом Министерства образования и науки Российской Федерации от 1 июля 2013 г. № 499»; приказа Министерства образования и науки РФ от 23 августа 2017 г. № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»; паспорта федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика Российской Федерации»; постановления Правительства Российской Федерации от 13 мая 2021 г. № 729 «О мерах по реализации программы стратегического лидерства «Приоритет-2030» (в редакции постановления Правительства Российской Федерации от 14 марта 2022 г. № 357 «О внесении изменений в постановление Правительства Российской Федерации от 13 мая 2021 г. № 729»); приказа Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 28 февраля 2022 г. № 143 «Об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации» и признании утратившими силу некоторых приказов

Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации об утверждении методик расчета показателей федеральных проектов национальной программы «Цифровая экономика Российской Федерации»; федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.02 Информационные системы и технологии (уровень бакалавриата), утвержденного приказом Минобрнауки России от 19 сентября 2017 г. № 926, (далее — ФГОС ВО), а также профессионального стандарта 06.001 «Программист», утвержденного приказом Министерства труда и социальной защиты РФ от 18 ноября 2013 г. № 679н.

Профессиональная переподготовка заинтересованных лиц (далее — Слушатели), осуществляемая в соответствии с Программой, имеющей отраслевую направленность «Экономика, финансы и управление», проводится в ФГАОУ ВО «Сибирский федеральный университет» (далее — Университет) в соответствии с учебным планом в очно-заочной форме обучения.

Разделы, включенные в учебный план Программы, используются для последующей разработки календарного учебного графика, учебно-тематического плана, рабочих программ модулей (дисциплин), оценочных и методических материалов. Перечисленные документы разрабатываются Университетом самостоятельно, с учетом актуальных положений законодательства об образовании, законодательства в области информационных технологий и смежных областей знаний ФГОС ВО и профессионального стандарта 06.001 «Программист».

1.2. Цель программы

Целью ДПП ПП является формирование у слушателей, обучающихся по ОП ВО – программам бакалавриата и программам специалитета (начиная со 2 курса), программам магистратуры (начиная с 1 курса) по специальностям и направлениям подготовки, относящимся к отрасли «Экономика, финансы и управление» цифровых компетенций в области использования кроссплатформенных, гибридных и веб- технологии в мобильной разработке, методологии разработки производительных мобильных приложений и инструментов профилирования, отладки и тестирования, применения языков программирования для решения профессиональных задач, а также приобретение по итогам прохождения ДПП ПП новой квалификации «Программист».

1.3 Требования к поступающим

К обучению по Программе допускаются обучающиеся за счет бюджетных средств или по договорам об оказании платных образовательных услуг, освоившие программы бакалавриата в объеме не менее 1 курса (бакалавры 2 курса) по специальностям и направлениям подготовки, относящимся к экономической сфере.

1.4. Характеристика новой квалификации и связанных с ней видов профессиональной деятельности, трудовых функций и(или) уровней квалификации

1.4.1. Область профессиональной деятельности слушателя, прошедшего обучение по программе профессиональной переподготовки, в которой может осуществлять профессиональную деятельность: разработка программного обеспечения.

Выпускники могут осуществлять профессиональную деятельность в других областях и(или) сферах профессиональной деятельности при условии соответствия уровня их образования и полученных компетенций требованиям к квалификации работника.

1.4.2. Объекты профессиональной деятельности: разработка мобильных приложений.

Виды профессиональной деятельности: разработка программного обеспечения

1.4.3. Уровень квалификации. В соответствии с приказом Министерства труда и социальной защиты Российской Федерации от 20 июля 2022 г. № 424н «Об утверждении Профессионального стандарта «Программист», дополнительная профессиональная программа профессиональной переподготовки «Мобильные приложения для электронной торговли: основы фреймворка Flutter» обеспечивает достижение шестого уровня квалификации.

1.4.4. Компетенции (трудовые функции) в соответствии с профессиональным стандартом (формирование новых или совершенствование имеющихся)

Программа разработана в соответствии с актуальными квалификационными требованиями, профессиональными стандартами специалистов. Виды профессиональной деятельности, трудовые функции, указанные в профессиональном стандарте 06.001 «Программист», представлены в таблицах 1–2.

Характеристика новой квалификации, связанной с видом профессиональной деятельности и трудовыми функциями в соответствии с профессиональным стандартом 06.001 «Программист»

Трудовые действия	Трудовая функция	Обобщенная трудовая функция	Вид профессиональной деятельности
Составление формализованных описаний решений, поставленных задач в соответствии с требованиями технического задания или внутренних документов организации. Разработка алгоритмов решения поставленных задач в соответствии с требованиями технического задания или внутренних документов организации.	А/01.3 Формализация и алгоритмизация поставленных задач для разработки программного кода	А Разработка и отладка программного кода	Разработка программного обеспечения
Создание программного кода в соответствии с техническим заданием (готовыми спецификациями)	А/02.3 Написание программного кода с использованием языков программирования, определения и манипулирования данными в базах данных		
Проверка работоспособности компьютерного программного обеспечения на основе разработанных тестовых наборов данных	В/03.4 Проверка работоспособности программного обеспечения	В Проверка работоспособности и рефакторинг кода программного обеспечения	
Разработка, изменение архитектуры компьютерного программного обеспечения и ее согласование с системным аналитиком и архитектором программного обеспечения	D/03.6 Проектирование компьютерного программного обеспечения	D Разработка требований и проектирование программного обеспечения	

Таблица 2

Характеристика новой и развиваемой цифровой компетенции в ИТ-сфере, связанной с уровнем формирования и развития в результате освоения программы «Мобильные приложения для электронной торговли: основы фреймворка Flutter»

Наименование сферы	Тип компетенции	Наименование компетенции	Номер компетенции (ID)	БАЗОВЫЙ УРОВЕНЬ РАЗВИТИЯ КОМПЕТЕНЦИЙ Способность проявляется под внешним контролем / при внешней постановке задачи/ обучающийся пользуется готовыми, рекомендованными продуктами
Разработка мобильных приложений	Компетенция применима в различных отраслях экономики	Использует кроссплатформенные, гибридные и веб- технологии в мобильной разработке	213	Использует только одну технологическую платформу (и язык программирования), имеет опыт работы с несколькими библиотеками
Разработка мобильных приложений	Компетенция применима в различных отраслях экономики	Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования	214	Использует методологии и шаблоны построения приложений с высокой производительностью и качеством кода под внешним контролем
Средства программной разработки	Компетенция применима в различных	Применяет языки программирования для решения	28	Применяет языки программирования для решения профессиональных задач под контролем более опытных специалистов

	отраслях экономики	профессиональн ых задач		
--	-----------------------	----------------------------	--	--

Таблица 3

Структура образовательных результатов

ID и формулировка целевого уровня формирования компетенций	Промежуточные образовательные результаты		
	Опыт практической деятельности (ОПД)	Умения (У)	Знания (З)
(213) Использует кроссплатформенные, гибридные и веб-технологии в мобильной разработке	<p>Разработка мобильных приложений с использованием кроссплатформенного фреймворка Flutter для Android и iOS.</p> <p>Создание мобильных интерфейсов на основе виджетов Flutter; интеграция веб-</p>	<p>Умение создавать и настраивать мобильные приложения на Flutter.</p> <p>Способность использовать основные компоненты и виджеты для построения пользовательских интерфейсов.</p>	<p>Знание принципов и архитектуры кроссплатформенной мобильной разработки на Flutter.</p> <p>Понимание основных возможностей фреймворка и особенностей его</p>

	<p>сервисов и внешних API в мобильные приложения.</p> <p>Разработка и внедрение прогрессивных веб-приложений (PWA) на Flutter.</p> <p>Тестирование, оптимизация производительности и публикация кроссплатформенных мобильных приложений.</p>	<p>Навык интеграции мобильных приложений с внешними сервисами и API.</p> <p>Умение оптимизировать и адаптировать приложения под разные устройства и платформы.</p> <p>Владение методами тестирования и отладки кроссплатформенных решений.</p> <p>Способность готовить приложения к публикации в магазинах приложений.</p>	<p>использования для гибридной и веб-разработки.</p> <p>Знание способов интеграции web-технологий и API в мобильные приложения.</p> <p>Понимание различий между нативной, гибридной и кроссплатформенной разработкой.</p> <p>Знание методов оптимизации и обеспечения безопасности мобильных приложений.</p>
<p>(214) Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования</p>	<p>Применение современных методологий и подходов для разработки производительных мобильных приложений.</p> <p>Анализ и оптимизация производительности приложений с использованием инструментов профилирования (например, Flutter DevTools, Android Profiler, Xcode Instruments).</p> <p>Проведение отладки и устранение ошибок с помощью встроенных средств разработки.</p> <p>Выполнение модульного, интеграционного и автоматизированного тестирования мобильных приложений.</p> <p>Внедрение практик мониторинга и сбора аналитики для контроля качества и стабильности приложений.</p>	<p>Умение использовать инструменты профилирования для анализа и оптимизации производительности мобильных приложений.</p> <p>Способность выявлять и устранять узкие места, влияющие на скорость работы и потребление ресурсов.</p> <p>Навык эффективной отладки и поиска ошибок в коде с помощью встроенных средств разработки.</p> <p>Умение организовывать и проводить различные виды тестирования (юнит-тесты, интеграционные, функциональные тесты).</p> <p>Способность внедрять мониторинг и использовать аналитические данные для повышения стабильности и качества мобильных приложений.</p>	<p>Знание современных методологий и практик по обеспечению производительности мобильных приложений.</p> <p>Понимание принципов работы инструментов профилирования, отладки и тестирования для различных платформ.</p> <p>Знание подходов к оптимизации использования памяти, процессора и других ресурсов устройства.</p> <p>Знание особенностей организации тестирования мобильных приложений и инструментов автоматизации тестирования.</p> <p>Понимание важности мониторинга, логирования и анализа пользовательских сценариев для повышения качества мобильных приложений.</p>

<p>(28) Применяет языки программирования для решения профессиональных задач</p>	<p>Разработка программных решений с использованием языков программирования (например, Dart) для автоматизации профессиональных задач и создания эффективных программных продуктов.</p> <p>Проектирование и реализация алгоритмов, написание модулей и скриптов для обработки данных, интеграция с внешними сервисами и API.</p> <p>Разработка, отладка и сопровождение программных решений, обеспечивающих выполнение прикладных задач в профессиональной деятельности.</p> <p>Тестирование, рефакторинг и оптимизация кода для повышения производительности и надежности программных решений.</p>	<p>Умение применять языки программирования для разработки программных продуктов.</p> <p>Навык проектирования и реализации алгоритмов для решения профессиональных задач.</p> <p>Владение приемами структурирования, документирования и оптимизации кода.</p> <p>Умение интегрировать программные решения с внешними сервисами, базами данных и API.</p> <p>Способность проводить тестирование, отладку и сопровождение программных решений.</p>	<p>Знание синтаксиса и основных конструкций используемых языков программирования.</p> <p>Понимание принципов объектно-ориентированного, процедурного и функционального программирования.</p> <p>Знание методов интеграции программных продуктов с внешними ресурсами и сервисами.</p> <p>Понимание особенностей работы с различными типами данных и алгоритмами обработки информации.</p> <p>Знание методов тестирования, отладки и оптимизации программных решений.</p>
---	--	---	--

1.5. Требования к материально-техническому обеспечению, необходимому для реализации дополнительной профессиональной программы профессиональной переподготовки (требования к аудитории, компьютерному классу, программному обеспечению)

Обучение производится на платформе электронного обучения СФУ «е-Курсы» (<https://e.sfu-kras.ru/>). Используются сервисы вебинаров и видеоконференций.

При проведении лекций, практических занятий, самостоятельной работы слушателей и стажировки используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой, высокоскоростное подключение к Интернет (не менее 5 Мбит/с).

Программное обеспечение (обновленное до последней версии): Java, Dart, Flutter, Android Studio

1.6. Особенности (принципы) построения дополнительной профессиональной программы профессиональной переподготовки

Особенности построения программы переподготовки «Мобильные приложения для электронной торговли: основы фреймворка Flutter»:

- в основу проектирования программы положен компетентностный подход;
- выполнение учебных заданий, требующих практического применения знаний и умений, полученных в ходе изучения логически связанных дисциплин;
- выполнение итоговых аттестационных работ по реальному заданию;
- использование информационных и коммуникационных технологий, в том числе современных систем технологической поддержки процесса обучения, обеспечивающих комфортные условия для обучающихся, преподавателей;
- применение электронных образовательных ресурсов (дистанционное, электронное, комбинированное обучение и пр.).

В поддержку дополнительной профессиональной программы профессиональной переподготовки разработан электронный курс на платформе е-Курсов СФУ: <http://e.sfu-kras.ru>.

1.7. Особенности организации практики

Практика слушателей дополнительной профессиональной программы переподготовки «Мобильные приложения для электронной торговли: основы фреймворка Flutter» представляет собой вид учебной деятельности, непосредственно ориентированный на профессионально-практическую подготовку слушателей.

Практика осуществляется в целях формирования и закрепления профессиональных умений и навыков, полученных в результате теоретической подготовки.

Сроки проведения практики устанавливаются графиком учебного процесса в объеме 36 часов в конце процесса обучения в соответствии с утвержденным в установленном порядке учебно-тематическим планом.

В рамках очно-заочной формы обучения на основе дистанционных технологий практика осуществляется в форме online практики.

1.8. Документ об образовании: диплом о переподготовке установленного образца.

УЧЕБНЫЙ ПЛАН
дополнительной профессиональной программы профессиональной переподготовки
«Мобильные приложения для электронной торговли: основы фреймворка Flutter»

Форма обучения – очно-заочная.

Срок обучения – 256 часов.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч	Формы контроля
				Лекции	Лабораторные работы	Практические и семинарские занятия		
1.	Основы языка программирования Dart	36	18	6		12	18	Зачет
2.	Основы фреймворка Flutter	54	28	10		18	26	Зачет
3.	Работа с данными на Dart и Flutter	54	28	10		18	26	Зачет
4.	Работа с платформой	26	12	6		6	14	Зачет
5.	Анимация и тестирование	26	12	6		6	14	Зачет
6.	Практика	36	18	-		18	18	Зачет
7.	Итоговая аттестация	24	8	-		8	16	Защита итоговой аттестационной работы (проекта)
	Итого	256	124	38		86	132	

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
дополнительной профессиональной программы профессиональной переподготовки
«Мобильные приложения для электронной торговли: основы фреймворка Flutter»

Категория слушателей: лица, имеющие или получающие высшее образование.

Срок обучения: 256 часов.

Форма обучения: очно-заочная.

Режим занятий: 6 часов в неделю.

№ п/п	Наименование дисциплин	Общая трудоемкость, ч	Всего контактн., ч	Контактные часы			СРС, ч
				Лекции	Лабораторные работы	Практ. и семинарские занятия	
1	Основы языка программирования Dart	36	18	6		12	18
2.1	Введение в Dart	8	4	2		2	4
2.2	Null-safety	8	4	2		2	4
2.3	ООП в Dart	8	4	2		2	4
2.4	Concurrency + Streams	12	6	2		4	6
2	Основы фреймворка Flutter	54	28	10		18	26
2.1	Что такое Flutter	14	8	2		6	6
2.2	Как устроен Flutter	20	10	4		6	10
2.3	Простые интерфейсы на Flutter	20	10	4		6	10
3	Работа с данными на Dart и Flutter	54	28	10		18	26
3.1	Кодогенерация в Dart	12	6	2		4	6
3.2	Управление состоянием приложения	12	6	2		4	6
3.3	Сериализация и десериализация JSON	8	4	2		2	4
3.4	Работа с сетью (REST API, WebSockets)	12	6	2		4	6
3.5	Работа с локальными данными	8	4	2		2	4
4	Работа с платформой	26	12	6		6	14
4.1	Введение в iOS и Android	10	4	2		2	6
4.2	Как Flutter работает с платформенным кодом	16	8	4		4	8
5	Анимация и тестирование	26	12	6		6	14
5.1	Анимации в Flutter	10	4	2		2	6

№ п/п	Наименование дисциплин	Общая трудоем- кость, ч	Всего контактн., ч	Контактные часы			СРС, ч
				Лекции	Лабораторные работы	Практ. и семинарские занятия	
5.2	Тестирование Flutter приложений	16	8	4		4	8
6	Практика	36	18			18	18
7	Итоговая аттестация	24	8	-		8	16
	Всего	256	128	32		96	128

II. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ

2.1. Формы аттестации, оценочные материалы, методические материалы

Программа предусматривает проведение текущей и итоговой аттестации. Текущая аттестация слушателей проводится по дисциплинам на основе выполнения заданий в электронном обучающем курсе, а также с учетом результатов промежуточного ассесмента.

Методические материалы, необходимые для выполнения текущих заданий, представлены в соответствующих элементах электронного обучающего курса и включают описание задания, методические рекомендации по его выполнению, критерии оценивания.

2.2. Требования и содержание итоговой аттестации

После завершения обучения по Программе обучающиеся допускаются к итоговой аттестации. Аттестация проводится с участием представителей профильных индустриальных партнеров. Итоговая аттестация по программе включает выполнение итоговой аттестационной работы (ИАР) в форме проекта. Основная цель итоговой аттестационной работы— выполнить работу, демонстрирующую уровень подготовленности к самостоятельной профессиональной деятельности.

ИАР выполняется индивидуально или в группах по 2-4 человека. Слушатель предоставляет результат выполненной работы в формате PDF, оформленной и отвечающей требованиям к содержанию итоговой аттестационной работы. Список использованных источников литературы приводится в конце ИАР. Документ прикрепляется в организационный электронный курс программы профессиональной переподготовки. В итоговой аттестационной работе должны быть четко обозначены область и актуальность работы, постановка задачи, приведены результаты, полученные слушателем. Требования и содержание итоговой аттестации изложены в методических указаниях к выполнению ИАР и размещаются на платформе электронных курсов СФУ.

Выполнение итоговой аттестационной работы является обязательным.

По результатам выполнения ИАР аттестационная комиссия принимает решение о предоставлении слушателям по результатам освоения дополнительной профессиональной программы профессиональной переподготовки права заниматься профессиональной деятельностью в сфере разработки программного обеспечения и выдаче диплома о профессиональной переподготовке.

Примерные задания на выполнение ИАР.

Вариант 1: Разработка мобильного приложения интернет-магазина (на примере e-commerce платформы)

Описание разработки:

Студенты должны разработать мобильное приложение интернет-магазина на Flutter, позволяющее пользователям просматривать каталог товаров, добавлять товары в корзину и оформлять заказы. Приложение должно взаимодействовать с серверной частью (REST API), обеспечивать авторизацию пользователей и визуализировать основные метрики магазина (например, график продаж, активность пользователей).

Компетенции, оцениваемые в задании:

1. Использует кроссплатформенные, гибридные и веб- технологии в мобильной разработке	
Показатели оценки	Оценка/норма
Корректность кроссплатформенной реализации	<p>Оценка: Приложение одинаково функционирует на Android и iOS, реализовано в виде PWA при необходимости.</p> <p>Норма: Отсутствие критических багов и несовместимостей; пользовательский сценарий полностью повторяется на обеих платформах и в веб-версии.</p>
Использование компонентов Flutter и веб-технологий	<p>Оценка: В приложении использованы стандартные и кастомные виджеты, реализованы responsive-интерфейсы, при необходимости — интеграция WebView или других веб-технологий.</p> <p>Норма: Интерфейс адаптируется под разные устройства (разрешения, ориентация экрана), все элементы корректно отображаются и работают.</p>
Интеграция с внешними API и сервисами	<p>Оценка: Реализована устойчивая интеграция с REST API, сторонними сервисами (например, аутентификация через OAuth, push-уведомления и т.д.)</p> <p>Норма: Все сервисы корректно взаимодействуют с приложением, ошибки обработки исключений минимизированы.</p>
Готовность к публикации	<p>Оценка: Приложение собрано и протестировано для Google Play, App Store и (если требуется) как PWA.</p>

	<p>Норма: Прохождение всех стандартных проверок и требований платформ, наличие необходимых файлов (manifest, assets, splash screen и пр.).</p>
<p>2. Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования</p>	
Показатели оценки	Оценка/норма
Оптимизация производительности	<p>Оценка: Использованы инструменты профилирования (Flutter DevTools, Android Profiler), проведён анализ производительности (анализ времени отклика, потребления памяти, FPS).</p> <p>Норма: Время отклика интерфейса — не более 1,5-2 секунд; потребление памяти и ресурсов соответствует типовым приложениям данной категории.</p>
Проведение тестирования	<p>Оценка: Реализованы модульные, интеграционные и/или UI-тесты с использованием соответствующих инструментов (Flutter test, Mockito, Espresso и т.д.)</p> <p>Норма: Покрытие кода тестами не менее 60%; приложение успешно проходит основные пользовательские сценарии в автоматизированных тестах.</p>
Использование методологий разработки	<p>Оценка: Применены современные подходы (например, Clean Architecture, MVVM, BLoC, SOLID-принципы) при проектировании приложения; есть документация по архитектуре.</p> <p>Норма: Архитектурные решения обоснованы и зафиксированы в проектной документации или README.</p>
Отладка и устранение ошибок	<p>Оценка: Использованы инструменты логирования, отладки (debug console, breakpoints, error handling).</p>

	<p>Норма: В процессе тестирования выявленные ошибки зафиксированы и устранены, в приложении реализована обработка исключительных ситуаций.</p>
3. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность выполнения программы	<p>Оценка: приложение корректно реализует все функции: отображение каталога, корзина, оформление заказа, авторизация.</p> <p>Норма: пользовательский сценарий проходит без сбоев, заказы формируются и отправляются на сервер.</p>
Эффективность кода	<p>Оценка: код написан с использованием эффективных практик Flutter и Dart (State Management, оптимизация сетевых запросов).</p> <p>Норма: время отклика основных функций приложения — не более 2 секунд.</p>
Документация кода	<p>Оценка: код содержит комментарии и документацию.</p> <p>Норма: каждый крупный виджет и сервис имеет docstring; README с инструкцией по запуску.</p>
Обоснованность решений	<p>Оценка: выбор архитектуры (например, BLoC, Provider, MVC) обоснован и описан</p> <p>Норма: в конце проекта приложен краткий отчет по архитектуре и интеграции с сервером (API).</p>

Вариант 2: Анализ пользовательского поведения и продаж в мобильном приложении

Описание разработки:

Создание мобильного приложения на Flutter, которое загружает данные о заказах и активности пользователей из CSV-файла, обрабатывает их (с использованием пакетов для работы с данными, например, csv, collection) и визуализирует результаты (например, график количества заказов по дням, активности пользователей, средний чек). Студенты должны проанализировать данные и предоставить выводы по эффективности продаж и пользовательскому поведению..

Компетенции, оцениваемые в задании:

1. Использует кроссплатформенные, гибридные и веб- технологии в мобильной разработке	
Показатели оценки	Оценка/норма
Корректность кроссплатформенной реализации	<p>Оценка: Приложение одинаково функционирует на Android и iOS, реализовано в виде PWA при необходимости.</p> <p>Норма: Отсутствие критических багов и несовместимостей; пользовательский сценарий полностью повторяется на обеих платформах и в веб-версии.</p>
Использование компонентов Flutter и веб-технологий	<p>Оценка: В приложении использованы стандартные и кастомные виджеты, реализованы responsive-интерфейсы, при необходимости — интеграция WebView или других веб-технологий.</p> <p>Норма: Интерфейс адаптируется под разные устройства (разрешения, ориентация экрана), все элементы корректно отображаются и работают.</p>
Интеграция с внешними API и сервисами	<p>Оценка: Реализована устойчивая интеграция с REST API, сторонними сервисами (например, аутентификация через OAuth, push-уведомления и т.д.)</p> <p>Норма: Все сервисы корректно взаимодействуют с приложением, ошибки обработки исключений минимизированы.</p>

Готовность к публикации	<p>Оценка: Приложение собрано и протестировано для Google Play, App Store и (если требуется) как PWA.</p> <p>Норма: Прохождение всех стандартных проверок и требований платформ, наличие необходимых файлов (manifest, assets, splash screen и пр.).</p>
2. Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования	
Показатели оценки	Оценка/норма
Оптимизация производительности	<p>Оценка: Использованы инструменты профилирования (Flutter DevTools, Android Profiler), проведён анализ производительности (анализ времени отклика, потребления памяти, FPS).</p> <p>Норма: Время отклика интерфейса — не более 1,5-2 секунд; потребление памяти и ресурсов соответствует типовым приложениям данной категории.</p>
Проведение тестирования	<p>Оценка: Реализованы модульные, интеграционные и/или UI-тесты с использованием соответствующих инструментов (Flutter test, Mockito, Espresso и т.д.)</p> <p>Норма: Покрытие кода тестами не менее 60%; приложение успешно проходит основные пользовательские сценарии в автоматизированных тестах.</p>
Использование методологий разработки	<p>Оценка: Применены современные подходы (например, Clean Architecture, MVVM, BLoC, SOLID-принципы) при проектировании приложения; есть документация по архитектуре.</p> <p>Норма: Архитектурные решения обоснованы и зафиксированы в проектной документации или README.</p>

Отладка и устранение ошибок	<p>Оценка: Используются инструменты логирования, отладки (debug console, breakpoints, error handling).</p> <p>Норма: В процессе тестирования выявленные ошибки зафиксированы и устранены, в приложении реализована обработка исключительных ситуаций.</p>
3. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность выполнения программы	<p>Оценка: приложение корректно загружает и обрабатывает данные из CSV (например, с помощью пакета csv для Flutter).</p> <p>Норма: результаты анализа должны соответствовать данным из CSV файла.</p>
Эффективность кода	<p>Оценка: код написан с использованием оптимальных методов обработки данных.</p> <p>Норма: время выполнения обработки данных не превышает 10 секунд для 1000 записей.</p>
Документация кода	<p>Оценка: код содержит комментарии и документацию.</p> <p>Норма: каждая функция имеет строку документации (docstring).</p>
Обоснованность решений	<p>Оценка: все методы анализа обоснованы.</p> <p>Норма: отчет должен содержать объяснение выбора методов анализа данных.</p>

Вариант 3: Разработка панели мониторинга для владельца интернет-магазина

Описание разработки:

Студенты должны создать мобильное приложение на Flutter с интерактивным графическим интерфейсом, который позволяет владельцу интернет-магазина в реальном времени просматривать ключевые метрики: количество заказов, средний чек, активных пользователей онлайн и продажи за выбранный период. Пользователь (владелец магазина) может выбирать параметры периода, фильтровать данные, и мгновенно видеть обновлённые графики и показатели. Компетенции, оцениваемые в задании:

1. Использует кроссплатформенные, гибридные и веб- технологии в мобильной разработке	
Показатели оценки	Оценка/норма
Корректность кроссплатформенной реализации	<p>Оценка: Приложение одинаково функционирует на Android и iOS, реализовано в виде PWA при необходимости.</p> <p>Норма: Отсутствие критических багов и несовместимостей; пользовательский сценарий полностью повторяется на обеих платформах и в веб-версии.</p>
Использование компонентов Flutter и веб-технологий	<p>Оценка: В приложении использованы стандартные и кастомные виджеты, реализованы responsive-интерфейсы, при необходимости — интеграция WebView или других веб-технологий.</p> <p>Норма: Интерфейс адаптируется под разные устройства (разрешения, ориентация экрана), все элементы корректно отображаются и работают.</p>
Интеграция с внешними API и сервисами	<p>Оценка: Реализована устойчивая интеграция с REST API, сторонними сервисами (например, аутентификация через OAuth, push-уведомления и т.д.)</p> <p>Норма: Все сервисы корректно взаимодействуют с приложением, ошибки обработки исключений минимизированы.</p>
Готовность к публикации	Оценка: Приложение собрано и протестировано для Google Play, App Store и (если требуется) как PWA.

	<p>Норма: Прохождение всех стандартных проверок и требований платформ, наличие необходимых файлов (manifest, assets, splash screen и пр.).</p>
<p>2. Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования</p>	
Показатели оценки	Оценка/норма
Оптимизация производительности	<p>Оценка: Использованы инструменты профилирования (Flutter DevTools, Android Profiler), проведён анализ производительности (анализ времени отклика, потребления памяти, FPS).</p> <p>Норма: Время отклика интерфейса — не более 1,5-2 секунд; потребление памяти и ресурсов соответствует типовым приложениям данной категории.</p>
Проведение тестирования	<p>Оценка: Реализованы модульные, интеграционные и/или UI-тесты с использованием соответствующих инструментов (Flutter test, Mockito, Espresso и т.д.)</p> <p>Норма: Покрытие кода тестами не менее 60%; приложение успешно проходит основные пользовательские сценарии в автоматизированных тестах.</p>
Использование методологий разработки	<p>Оценка: Применены современные подходы (например, Clean Architecture, MVVM, BLoC, SOLID-принципы) при проектировании приложения; есть документация по архитектуре.</p> <p>Норма: Архитектурные решения обоснованы и зафиксированы в проектной документации или README.</p>
Отладка и устранение ошибок	<p>Оценка: Использованы инструменты логирования, отладки (debug console, breakpoints, error handling).</p>

	<p>Норма: В процессе тестирования выявленные ошибки зафиксированы и устранены, в приложении реализована обработка исключительных ситуаций.</p>
<p>3. Применяет языки программирования для решения профессиональных задач</p>	
Показатели оценки	Оценка/норма
Корректность выполнения программы	<p>Оценка: реализован интуитивно понятный и функциональный интерфейс на Flutter для мониторинга метрик e-commerce в реальном времени.</p> <p>Норма: все функции интерфейса работают без ошибок.</p>
Эффективность кода	<p>Оценка: код написан с использованием оптимальных подходов к обработке событий.</p> <p>Норма: время отклика интерфейса не превышает 1 секунды.</p>
Документация кода	<p>Оценка: код содержит комментарии и документацию.</p> <p>Норма: каждая функция имеет строку документации (docstring).</p>
Обоснованность решений	<p>Оценка: все выбранные элементы интерфейса обоснованы.</p> <p>Норма: отчет должен содержать объяснение выбора элементов интерфейса.</p>

Вариант 4: Создание мобильного приложения для управления товарами и заказами

Описание разработки:

Разработка мобильного приложения на Flutter, которое позволяет сотрудникам интернет-магазина управлять данными о товарах и заказах. Пользователь может добавлять, редактировать и удалять товары, просматривать

и обновлять статусы заказов. Все изменения сохраняются в облачной базе данных (например, Firebase или Supabase), а актуальная информация отображается в приложении в реальном времени.

Компетенции, оцениваемые в задании:

1. Использует кроссплатформенные, гибридные и веб-технологии в мобильной разработке	
Показатели оценки	Оценка/норма
Корректность кроссплатформенной реализации	<p>Оценка: Приложение одинаково функционирует на Android и iOS, реализовано в виде PWA при необходимости.</p> <p>Норма: Отсутствие критических багов и несовместимостей; пользовательский сценарий полностью повторяется на обеих платформах и в веб-версии.</p>
Использование компонентов Flutter и веб-технологий	<p>Оценка: В приложении использованы стандартные и кастомные виджеты, реализованы responsive-интерфейсы, при необходимости — интеграция WebView или других веб-технологий.</p> <p>Норма: Интерфейс адаптируется под разные устройства (разрешения, ориентация экрана), все элементы корректно отображаются и работают.</p>
Интеграция с внешними API и сервисами	<p>Оценка: Реализована устойчивая интеграция с REST API, сторонними сервисами (например, аутентификация через OAuth, push-уведомления и т.д.)</p> <p>Норма: Все сервисы корректно взаимодействуют с приложением, ошибки обработки исключений минимизированы.</p>
Готовность к публикации	<p>Оценка: Приложение собрано и протестировано для Google Play, App Store и (если требуется) как PWA.</p> <p>Норма: Прохождение всех стандартных проверок и требований</p>

	платформ, наличие необходимых файлов (manifest, assets, splash screen и пр.).
2. Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования	
Показатели оценки	Оценка/норма
Оптимизация производительности	<p>Оценка: Использованы инструменты профилирования (Flutter DevTools, Android Profiler), проведён анализ производительности (анализ времени отклика, потребления памяти, FPS).</p> <p>Норма: Время отклика интерфейса — не более 1,5-2 секунд; потребление памяти и ресурсов соответствует типовым приложениям данной категории.</p>
Проведение тестирования	<p>Оценка: Реализованы модульные, интеграционные и/или UI-тесты с использованием соответствующих инструментов (Flutter test, Mockito, Espresso и т.д.)</p> <p>Норма: Покрытие кода тестами не менее 60%; приложение успешно проходит основные пользовательские сценарии в автоматизированных тестах.</p>
Использование методологий разработки	<p>Оценка: Применены современные подходы (например, Clean Architecture, MVVM, BLoC, SOLID-принципы) при проектировании приложения; есть документация по архитектуре.</p> <p>Норма: Архитектурные решения обоснованы и зафиксированы в проектной документации или README.</p>
Отладка и устранение ошибок	<p>Оценка: Использованы инструменты логирования, отладки (debug console, breakpoints, error handling).</p> <p>Норма: В процессе тестирования выявленные ошибки зафиксированы</p>

	и устранены, в приложении реализована обработка исключительных ситуаций.
3. Применяет языки программирования для решения профессиональных задач	
Показатели оценки	Оценка/норма
Корректность выполнения программы	Оценка: преализованы функции загрузки, добавления, редактирования и удаления товаров и заказов через интерфейс. Норма: все функции CRUD работают без ошибок.
Эффективность кода	Оценка: код написан с использованием оптимальных подходов к работе с базами данных. Норма: время обработки запросов не превышает 2 секунд.
Документация кода	Оценка: код содержит комментарии и документацию. Норма: каждая функция имеет строку документации (docstring).
Обоснованность решений	Оценка: все методы работы с данными обоснованы. Норма: отчет должен содержать объяснение выбора технологий для разработки.

Вариант 5. Проектирование и интеграция базы данных для автоматизации управления интернет-магазином

Описание задания:

Студенты разрабатывают мобильное приложение на Flutter, интегрированное с собственной реляционной базой данных (например, SQLite или облачной СУБД), которая автоматизирует процессы управления каталогом товаров, клиентами и заказами. Приложение должно обеспечивать хранение, быстрый поиск и фильтрацию данных, а также поддерживать офлайн-доступ к основной информации. Компетенции, оцениваемые в задании:

1. Использует кроссплатформенные, гибридные и веб- технологии в мобильной разработке	
Показатели оценки	Оценка/норма
Корректность кроссплатформенной реализации	<p>Оценка: Приложение одинаково функционирует на Android и iOS, реализовано в виде PWA при необходимости.</p> <p>Норма: Отсутствие критических багов и несовместимостей; пользовательский сценарий полностью повторяется на обеих платформах и в веб-версии.</p>
Использование компонентов Flutter и веб-технологий	<p>Оценка: В приложении использованы стандартные и кастомные виджеты, реализованы responsive-интерфейсы, при необходимости — интеграция WebView или других веб-технологий.</p> <p>Норма: Интерфейс адаптируется под разные устройства (разрешения, ориентация экрана), все элементы корректно отображаются и работают.</p>
Интеграция с внешними API и сервисами	<p>Оценка: Реализована устойчивая интеграция с REST API, сторонними сервисами (например, аутентификация через OAuth, push-уведомления и т.д.)</p> <p>Норма: Все сервисы корректно взаимодействуют с приложением, ошибки обработки исключений минимизированы.</p>
Готовность к публикации	<p>Оценка: Приложение собрано и протестировано для Google Play, App Store и (если требуется) как PWA.</p> <p>Норма: Прохождение всех стандартных проверок и требований платформ, наличие необходимых файлов (manifest, assets, splash screen и пр.).</p>
2. Использует методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования	

Показатели оценки	Оценка/норма
Оптимизация производительности	<p>Оценка: Использованы инструменты профилирования (Flutter DevTools, Android Profiler), проведён анализ производительности (анализ времени отклика, потребления памяти, FPS).</p> <p>Норма: Время отклика интерфейса — не более 1,5-2 секунд; потребление памяти и ресурсов соответствует типовым приложениям данной категории.</p>
Проведение тестирования	<p>Оценка: Реализованы модульные, интеграционные и/или UI-тесты с использованием соответствующих инструментов (Flutter test, Mockito, Espresso и т.д.)</p> <p>Норма: Покрытие кода тестами не менее 60%; приложение успешно проходит основные пользовательские сценарии в автоматизированных тестах.</p>
Использование методологий разработки	<p>Оценка: Применены современные подходы (например, Clean Architecture, MVVM, BLoC, SOLID-принципы) при проектировании приложения; есть документация по архитектуре.</p> <p>Норма: Архитектурные решения обоснованы и зафиксированы в проектной документации или README.</p>
Отладка и устранение ошибок	<p>Оценка: Использованы инструменты логирования, отладки (debug console, breakpoints, error handling).</p> <p>Норма: В процессе тестирования выявленные ошибки зафиксированы и устранены, в приложении реализована обработка исключительных ситуаций.</p>
3. Применяет языки программирования для решения профессиональных задач	

Показатели оценки	Оценка/норма
Корректность работы базы данных	<p>Оценка: приложение позволяет просматривать, добавлять, изменять и удалять записи; данные обновляются локально и (опционально) синхронизируются с сервером.</p> <p>Норма: все функции работают корректно и без ошибок, пользователь может легко взаимодействовать с базой данных.</p>
Эффективность кода	<p>Оценка: код написан с использованием оптимальных алгоритмов и структур данных.</p> <p>Норма: время выполнения операций (например, поиск материала) не превышает установленные пределы (например, менее 2 секунд на поиск).</p>
Документация кода	<p>Оценка: код должен содержать комментарии и документацию, объясняющую логику работы функций. Норма: Каждая функция имеет строку документации (docstring), описывающую ее назначение и параметры.</p>
Обоснованность решений	<p>Оценка: все выбранные методы и подходы должны быть обоснованы.</p> <p>Норма: в конце проекта должен быть представлен отчет, объясняющий выбор технологий и инструментов для разработки базы данных</p>

Эти темы могут быть адаптированы в зависимости от интересов студентов и специфики их обучения. Обучающиеся предоставляется право самостоятельно определить тему ИАР. Приветствуется выбор темы итогового проекта, связанный с профессиональной деятельностью студента по основному направлению подготовки или специальности.

Структура и содержание итоговой аттестационной работы

Задачи итоговой аттестационной работы должны иметь практическое значение. Во введении ИАР должна быть обоснована актуальность работы и сущность исследуемой проблемы, раскрыты цель, задачи, объект и предмет проектирования (разработки), методы выполнения проектирования (разработки). Как правило, ИАР состоит из 3 разделов (глав).

Глава 1 содержит обзор литературы по теме работы, в котором должны быть освещены различные точки зрения по затронутым в работе вопросам и обязательно сформулировано авторское отношение к ним, характеристика объекта и предмета разработки.

Глава 2 содержит характеристику методов выполнения работы, а также техническое задание на разработку программного продукта, включая перечень функций и возможностей, которые должен выполнять продукт, описание пользовательских сценариев и требований к интерфейсу, нефункциональные требования, этапы разработки и сроки их выполнения, описание методов и подходов к тестированию.

Глава 3 содержит характеристику результатов выполнения работы и их интерпретацию. В третьей главе студенту необходимо привести значимые участки программного кода, а также продемонстрировать визуально сценарии работы пользователя с программным продуктом.

В заключении формулируются конкретные выводы по работе и предложения по их реализации.

Этапы выполнения ИАР задаются графиком выполнения и контролируются руководителем. Примерный список основных этапов выполнения проекта представлен ниже:

1. Выбор темы ИАР. В начале обучения за студентом закрепляется руководитель итоговой работы, который прорабатывает тему. При выборе темы итогового проекта берется во внимание вопросы, связанные с профессиональной деятельностью студента по основному направлению подготовки или специальности.
2. Формулировка цели и задач ИАР работы. Обоснование актуальности выбранного направления работы.
3. Поиск аналогичных решений, формирование концепции предлагаемых подходов, методов.
4. Выбор и/или проектирование программного, технического и иных видов обеспечений для достижения цели и решения задач ИАР.
5. Разработка программной реализации проекта, проведение практических испытаний и анализ результатов.
6. Формирование итоговой аттестационной работы.
7. Представление ИАР аттестационной комиссии.

Критерии оценивания итоговой аттестационной работы

Слушатель получает оценку «удовлетворительно», если:

1. Контроль со стороны преподавателя:

- Разрабатывает программное обеспечение под контролем преподавателя, не проявляя инициативы в проектировании.
 - Выполняет задания по шаблонам или инструкциям без глубокого понимания процесса разработки.
2. Базовые навыки программирования:
- Демонстрирует базовые навыки программирования на Python, но испытывает трудности с более сложными задачами.
 - Использует простые конструкции языка, не применяя более продвинутые возможности.
3. Использование интегрированной среды разработки:
- Студент использует IDE, но демонстрирует ограниченные навыки работы с инструментами отладки и тестирования.
 - Проект имеет базовую структуру, однако не все файлы и каталоги организованы логически.
4. Корректность выполнения программы:
- Программа выполняет основные функции, однако могут быть ошибки в расчетах или выводе результатов.
 - Результаты не всегда соответствуют ожидаемым значениям по заданным параметрам.
5. Эффективность кода:
- Код работает, но может быть неэффективным и содержать излишние вычисления.
 - Код может содержать значительные недочеты, проблемы с читаемостью и структурой.
6. Документация кода:
- Код не содержит комментариев.
 - Документация отсутствует или крайне ограничена.
7. Обоснованность решений:
- В отчете представлены общие соображения, но недостаточно обоснований для выбора конкретных методов.

Слушатель получает оценку «хорошо», если:

1. Самостоятельная разработка модулей:
- Разрабатывает отдельные модули или компоненты программы с минимальным контролем со стороны преподавателя.
 - Умеет интегрировать свои модули в общую архитектуру проекта.
2. Использование интегрированной среды разработки:
- Студент уверенно использует IDE, применяет некоторые функции отладки и тестирования.
 - Проект структурирован логично, но может быть небольшое количество недочетов в организации файлов.
3. Корректность выполнения программы:
- Программа корректно выполняет заявленный функционал, хотя могут быть незначительные неточности в выводе результатов.

- Результаты в большинстве случаев соответствуют ожидаемым значениям.
4. Эффективность кода:
 - Код написан достаточно эффективно, однако есть возможности для оптимизации.
 - Время выполнения программы не превышает 5 секунд.
 5. Документация кода:
 - Код содержит комментарии и документацию, но некоторые функции могут быть недостаточно описаны.
 6. Обоснованность решений:
 - В отчете представлены обоснования выбора технологий, однако некоторые аспекты требуют более глубокого анализа.
 7. Работа в команде:
 - Участвует в командной разработке, взаимодействует с другими участниками проекта, но требует некоторого контроля.

Слушатель получает оценку «Отлично», если

1. Экспертное применение Flutter:
 - Применяет язык программирования Flutter на высоком уровне, демонстрируя глубокое понимание языка и его возможностей.
 - Использует продвинутые конструкции языка, такие как генераторы, декораторы и контекстные менеджеры.
2. Системный подход к разработке:
 - Контролирует весь цикл разработки программного обеспечения: от сбора требований до тестирования и развертывания.
 - Самостоятельно проводит анализ требований, проектирует архитектуру приложения и реализует ее.
3. Использование интегрированной среды разработки:
 - Студент демонстрирует отличные навыки работы с IDE, эффективно использует инструменты отладки и тестирования.
 - Проект имеет четкую и логичную структуру с хорошо организованными файлами и каталогами.
4. Корректность выполнения программы:
 - Программа полностью корректно выполняет заявленный функционал, результаты точно соответствуют ожидаемым значениям.
5. Эффективность кода:
 - Код написан оптимально, использует эффективные алгоритмы и структуры данных.
6. Документация кода:
 - Код содержит подробные комментарии.
7. Обоснованность решений:
 - В отчете представлено глубокое обоснование выбора методов и подходов с учетом различных факторов и альтернативных подходов.

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Основы языка программирования Dart»

1. Аннотация

В рамках дисциплины «Основы языка программирования Dart» слушатели освоят синтаксис языка программирования Dart, null-safety, реализацию объектно-ориентированного программирования в Dart, принципы применения многопоточного и асинхронного программирования в Dart.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:

PO1. Применять принципы и основы алгоритмизации

PO2. Применять выбранные языки программирования для написания программного кода

PO3. Разрабатывать процедуры проверки работоспособности программного обеспечения

PO4. Использовать существующие типовые решения и шаблоны проектирования программного обеспечения

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 1. Основы языка программирования Dart (36 часов)			
Введение в Dart (8 ч)	Синтаксис языка программирования Dart. Основные языковые конструкции (2 ч)	Операторы в Dart (2 х)	Коллекции, обработка исключений (4 ч)
Null-safety (8 ч)	Null-safety. Синтаксис и ключевые слова (2 ч)	Система типов в Dart с введением null safety (2 ч)	Функциональное программирование (4 ч)
ООП в Dart (8 ч)	Классы и объекты. Взаимоотношения между классами (2 ч)	Создание классов в Dart (2 ч)	Темы для самостоятельного изучения (4 ч)
Concurrency + Streams (16 ч.)	Многопоточное и асинхронное программирование (4 ч)	Применение Concurrency Streams (4 ч)	Темы для самостоятельного изучения (4 ч)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, Dart, Flutter, Android Studio.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Алеев, Андрей. *Быстрый старт Flutter-разработчика*. Litres, 2022.
2. Шингерей А. Н. *Разработка мобильных приложений с использованием фреймворка Flutter на языке программирования DART*. – 2022.
3. Бакетт К. *Dart в действии*. – Litres, 2022.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

1. Основы языка программирования Dart. URL: <https://metanit.com/dart/tutorial>
2. Dart Codelabs. URL: <https://dart.dev/codelabs>

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

тесты самоконтроля к лекциям 40 %;

практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Примеры тестов для контроля знаний

1. Для замены какого языка был создан Dart?

a. TypeScript

b. Java

c. JavaScript

2. Какого ключевого слова нет в Dart?

a. this

b. class

c. interface

Типовое практическое задание

Задание 1. Разработка консольного приложения на языке программирования Dart.

Цель задания: разработать и реализовать консольное приложение с использованием языка программирования Dart.

Инструкция:

Шаг 1. Основываясь на материалах лекции и самостоятельной работы, создать проект консольного приложения Dart.

Шаг 2. Создать приложение и заполнить его в соответствии с требованиями к проекту, созданном техническим заданием.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом занятии.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Основы фреймворка Flutter»

1. Аннотация

В рамках дисциплины «Основы фреймворка Flutter» слушатели получают знания по разработке приложений на языке Dart с использованием фреймворка Flutter, научатся создавать простые интерфейсы на Flutter.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:
PO5. Использовать кроссплатформенные, гибридные и веб- технологии в мобильной разработке

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 2. Основы фреймворка Flutter (54 часа)			
Что такое Flutter (16 ч)	Что такое Flutter (2 ч)	Установка и запуск первого Flutter приложения (6 ч)	Темы для самостоятельного изучения (8 ч)
Как устроен Flutter (16 ч)	Как устроен Flutter (2 ч)	Верстка простого интерфейса (6 ч)	Темы для самостоятельного изучения (8 ч)
Простые интерфейсы на Flutter (20 ч)	Простые интерфейсы на Flutter (4 ч)	Разработка навигации в приложении (6 ч)	Темы для самостоятельного изучения (10 ч)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение

теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиоколонками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, Dart, Flutter, Android Studio.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Алеев, Андрей. Быстрый старт Flutter-разработчика. Litres, 2022.
2. Шингерей А. Н. Разработка мобильных приложений с использованием фреймворка Flutter на языке программирования DART. – 2022.
3. Бакетт К. Dart в действии. – Litres, 2022.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

1. Building Layouts | Flutter. URL: <https://dart.dev/codelabs>

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

тесты самоконтроля к лекциям 40 %;

практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Примеры тестов для контроля знаний

1. Какое дерево отвечает за отрисовку пользовательского интерфейса?
 - a. Дерево виджетов
 - b. Дерево элементов
 - c. Дерево Render Objects

Типовое практическое задание

Задание 1. Создание проекта мобильного приложения на Flutter

Цель задания: создать и сверстать мобильное приложение с использованием фреймворка Flutter

Инструкция:

Шаг 1. Основываясь на материалах лекции и самостоятельной работы, создать проект Flutter.

Шаг 2. Создать пользовательский интерфейс приложения и заполнить его в соответствии с требованиями, созданными техническим заданием.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом занятии.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Работа с данными на Dart и Flutter»

1. Аннотация

В рамках дисциплины «Работа с данными на Dart и Flutter» слушатели получают знания о кодогенерации в Dart, получают навыки разработки модели данных, научатся принципам управления состоянием приложения, а также получать знания и практические навыки работы с сетью (REST API, WebSockets).

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:
 PO5. Использовать кроссплатформенные, гибридные и веб- технологии в мобильной разработке
 PO6. Использовать Android средства разработки.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 3. Работа с данными на Dart и Flutter (54 часа)			
Кодогенерация в Dart (12 ч)	Кодогенерация в Dart (2 ч)	Разработка модели данных (4 ч)	Темы для самостоятельного изучения (6 ч)
Управление состоянием приложения (12 ч)	Управление состоянием приложения (2 ч)	Управление состоянием приложения (4 ч)	Темы для самостоятельного изучения (6 ч)
Сериализация и десериализация JSON (8 ч)	Сериализация и десериализация JSON (2 ч)	Сериализация и десериализация JSON (2 ч)	Темы для самостоятельного изучения (4 ч)
Работа с сетью (REST API, WebSockets) (8 ч)	Работа с сетью (REST API, WebSockets) (2 ч)	Работа с сетью через REST API (2 ч)	Темы для самостоятельного изучения (4 ч)
Работа с локальными данными (8 ч)	Работа с локальными данными (2 ч)	Работа с key-value хранилищами (2 ч) Работа с базой данных (2 ч)	Темы для самостоятельного изучения (4 ч)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, Dart, Flutter, Android Studio.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. *Алеев, Андрей. Быстрый старт Flutter-разработчика. Litres, 2022.*
2. *Шингерей А. Н. Разработка мобильных приложений с использованием фреймворка Flutter на языке программирования DART. – 2022.*
3. *Бакетт К. Dart в действии. – Litres, 2022.*

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

тесты самоконтроля к лекциям 40 %;

практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Примеры тестов для контроля знаний

1. Как расшифровывается BLoC?
 - a. Business Logic Component
 - b. Business Logic Contruction
 - c. Build Legacy Code

Типовое практическое задание

Задание 1. Разработка мобильного приложения на Flutter

Цель задания: разработать и реализовать мобильное приложение с использованием фреймворка Flutter и библиотеки BLoC.

Инструкция:

Шаг 1. Создать проект мобильного приложения на фреймворке Flutter.

Шаг 2. Подключить рассмотренные в ходе лекций и самостоятельной работы библиотеки.

Шаг 3. Реализовать подключение к API в соответствии с требованиями.

Шаг 4. Реализовать в проекте пользовательский интерфейс вывода информации из API и отправке данных в API при помощи библиотеки BLoC.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом занятии.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА

дисциплины (модуля) «Работа с платформой»

1. Аннотация

В рамках дисциплины «Работа с платформой» слушатели получают знания о разработки мобильных приложений под операционными системами iOS и Android, а также о работе фреймворка Flutter с платформенным кодом.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:
PO5. Использовать кроссплатформенные, гибридные и веб- технологии в мобильной разработке

PO6. Использовать Android средства разработки.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 4. Работа с платформой (36 часов)			
Введение в iOS и Android (16 ч)	Введение в iOS и Android (2 ч)	Особенности разработки программного кода по ОС iOS и Android (6 ч)	Темы для самостоятельного изучения (8 ч)
Как Flutter работает с платформенным кодом (20 ч)	Как Flutter работает с платформенным кодом (4 ч)	Работа с платформенным кодом (6 ч)	Темы для самостоятельного изучения (10 ч)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение

теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, Dart, Flutter, Android Studio.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Алеев, Андрей. Быстрый старт Flutter-разработчика. Litres, 2022.
2. Шингерей А. Н. Разработка мобильных приложений с использованием фреймворка Flutter на языке программирования DART. – 2022.
3. Бакетт К. Dart в действии. – Litres, 2022.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Примеры тестов для контроля знаний

1. Какого типа общения с платформой нет в Flutter?
 - a. Method Channel
 - b. JavaScript Bridge
 - c. Event Channel

Типовое практическое задание

Задание 1. Ознакомление с подключением платформенного кода

Цель задания: разработать мобильное приложение с использованием фреймворка Flutter, часть кода которого выполняется нативно под Android.

Инструкция:

Шаг 1. Создать проект Flutter приложения.

Шаг 2. Основываясь на материалах лекции и самостоятельной работы, реализовать платформенный канал под Android.

Шаг 3. Реализовать функционал вызова функций через платформенный канал.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом занятии.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА
дисциплины (модуля)
«Анимация и тестирование»

1. Аннотация

В рамках дисциплины «Анимация и тестирование» слушатели получат знания тестировании Flutter приложений, в частности о unit-тестировании и о widget-тестировании, а также о использовании анимации в приложениях Flutter.

Цель дисциплины (результаты обучения)

По окончании обучения на данной дисциплине слушатели будут способны:
РО7. Использовать методологии разработки производительных мобильных приложений и инструменты профилирования, отладки и тестирования

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Модуль 4. Работа с платформой (36 часов)			
Анимации в Flutter (14 ч)	Анимации в Flutter (2 ч)	Анимации в Flutter (4 ч)	Темы для самостоятельного изучения (8 ч)
Тестирование Flutter приложений (22 ч)	Тестирование Flutter приложений (4 ч)	Unit и Widget тестирование Flutter приложений (8 ч)	Темы для самостоятельного изучения (8 ч)

3. Условия реализации программы дисциплины

Организационно-педагогические условия реализации программы

Обучение по программе реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Лекционный материал представляется в виде синхронных лекций, записей занятий, текстовых материалов, презентаций, размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение

теоретического материала (СРС) предполагается до и после синхронной части работы.

Материально-технические условия реализации программы

Синхронные занятия реализуются на базе инструментов видеоконференцсвязи и включают в себя лекционные и практические занятия. Для проведения синхронных занятий (вебинаров со спикерами) применяется программа видеоконференцсвязи. При проведении лекций, практических занятий, самостоятельной работы слушателей используется следующее оборудование: компьютер с наушниками или аудиокolonками, микрофоном и веб-камерой. Программное обеспечение (обновленное до последней версии): браузер Google Chrome, Java, Dart, Flutter, Android Studio.

Учебно-методическое и информационное обеспечение программы

Дисциплина может быть реализована как очно, так и заочно, в том числе, с применением дистанционных образовательных технологий. Она включает занятия лекционного типа, интерактивные формы обучения, практические занятия.

Содержание комплекта учебно-методических материалов

По данной дисциплине имеется электронный учебно-методический комплекс (УМК) в системе электронных курсов СФУ. УМК содержит: систему навигации по дисциплине (учебно-тематический план, интерактивный график работы по дисциплине, сведения о результатах обучения, чат для объявлений и вопросов преподавателю), текстовые материалы к лекциям, практические и тестовые задания, списки основной и дополнительной литературы. В электронном курсе реализована система обратной связи.

Литература

Основная литература

1. Алеев, Андрей. Быстрый старт Flutter-разработчика. Litres, 2022.
2. Шингерей А. Н. Разработка мобильных приложений с использованием фреймворка Flutter на языке программирования DART. – 2022.
3. Бакетт К. Dart в действии. – Litres, 2022.

Перечень ресурсов информационно-телекоммуникационной сети Интернет, необходимых для освоения дисциплины

4. Оценка качества освоения программы дисциплины (формы аттестации, оценочные и методические материалы)

Форма аттестации по дисциплине — зачет.

Оценка результатов обучения осуществляется следующим образом. Максимально за курс можно набрать 100%, из них:

- тесты самоконтроля к лекциям 40 %;
- практические задания составляют 60 %.

Зачет получают слушатели, набравшие не менее 50 % из 100% от общего прогресса по курсу.

Примеры тестов для контроля знаний

1. Какой mixin необходимо добавить к виджету для работы с анимациями вручную?
 - a. TickerProviderMixin
 - b. TickerMixin
 - c. AnimationMixin

Типовое практическое задание

Задание 1. Тестирование мобильного приложения на Flutter

Цель задания: Разработка Unit-тестов и Widget-тестов для мобильного приложения на Flutter.

Инструкция:

Шаг 1. Создать и реализовать приложение на Flutter, согласно требованиям.

Шаг 2. Основываясь на материалах лекции и самостоятельной работы, реализовать модульные тесты для проекта на фреймворке Flutter.

Шаг 3. Основываясь на материалах лекции и самостоятельной работы, реализовать автоматические тесты для компонентов пользовательского интерфейса для проекта на фреймворке Flutter.

Обратную связь по выполненной работе слушатель получит напрямую. Некоторые работы будут прокомментированы на практическом занятии.

Критерии оценивания заданий

Баллы	1 балл	2 балла	3 балла
Критерий	Задание выполнено частично, требует серьезной доработки	Задание выполнено, но требует некоторой доработки	Задание выполнено полностью, не требует доработки

РАБОЧАЯ ПРОГРАММА ПРАКТИКИ

1. Аннотация

Основная цель данной практики — обучить студентов основам программирования на Flutter и показать, как этот фреймворк может быть использован для создания приложений для электронной коммерции. Участники научатся собирать, обрабатывать и анализировать данные, а также разрабатывать небольшие программы для автоматизации процессов.

Практика включает в себя теоретические занятия, практические работы и проектную деятельность. Участники будут работать над реальными кейсами, что позволит им применить полученные знания на практике. Основной задачей практики является закрепление в практической деятельности профессиональных компетенций, умений, навыков и знаний, полученных в ходе обучения.

Планируемые результаты:

По окончании практики слушатели будут способны составлять и применять фреймворк Flutter для написания программного кода; осуществлять проверку работоспособности программного обеспечения на основе разработанных тестовых наборов данных; использовать при разработке программного обеспечения существующие типовые решения и шаблоны проектирования.

2. Содержание

№, наименование темы	Содержание лекций (кол-во часов)	Наименование практических (семинарских занятий) (кол-во часов)	Виды СРС (кол-во часов)
Практика (36 часов)			
1. Общие вопросы (ознакомление с предприятием)		Изучение общего технологического цикла производства программного обеспечения (4 ч.).	
2. Практическая часть		Решение практико-ориентированных задач. Обратная связь от экспертов (12 ч.)	Решение практико-ориентированных задач (18 ч.)

Содержание практики включает следующие этапы:

1. Знакомство с работодателями и презентация компаний:

На этом этапе представители e-commerce компаний расскажут о своих организациях, сферах деятельности и типичных задачах, которые они решают с помощью мобильных технологий. Участники смогут задать вопросы и узнать о реальных потребностях цифрового бизнеса в области мобильных приложений.

2. Обсуждение реальных кейсов:

Работодатели представят конкретные примеры задач, связанных с автоматизацией и оптимизацией онлайн-продаж, пользовательского опыта и управления заказами через мобильные приложения.

3. Формирование пула заданий:

Наставники и представители компаний определяют актуальные задачи, которые могут быть решены с помощью Flutter. Это могут быть задания по созданию интерфейсов интернет-магазинов, внедрению push-уведомлений, реализации аналитики пользовательского поведения, интеграции с платёжными системами и базами данных.

4. Выполнение проектных заданий:

Каждый участник выбирает одно или несколько заданий из сформированного пула, опираясь на свои интересы и навыки.

В процессе работы участники разрабатывают мобильное приложение, которое может включать в себя:

- Разработку и интеграцию пользовательских интерфейсов для e-commerce;
- Внедрение функций автоматизации бизнес-процессов (например, обработка заказов, управление каталогом товаров);
- Реализацию визуализации аналитических данных (продажи, активность пользователей и др.).

5. Проверка и тестирование:

После завершения работы над проектами участники демонстрируют свои приложения для проверки. Используются современные подходы к тестированию (unit-тесты, тестирование интерфейса), а также проводится проверка работоспособности приложений в различных сценариях, включая эмуляцию пользовательских действий.

6. Обратная связь и доработка проектов:

Участники получают комментарии и рекомендации от работодателей и наставников по своим приложениям. Обсуждаются сильные стороны и возможные направления для улучшения. На основе полученных замечаний студенты вносят доработки, улучшая качество и удобство использования своих мобильных решений.

3. Условия реализации программы практики

Организационные и педагогические условия реализации программы

Обучение по программе практики реализовано в формате смешанного обучения, с применением активных технологий совместного обучения в электронной среде (синхронные и асинхронные занятия). Материал практических занятий представляется в виде синхронных занятий, презентаций,

размещаемых в электронном курсе. Данные материалы сопровождаются заданиями и дискуссиями в чатах дисциплин. Изучение теоретического материала (СРС) предполагается до и после синхронной части работы.

Практика проводится под руководством назначенного руководителя из числа профессорско-преподавательского состава Университета, а также руководителя из состава организации, структурных подразделениях организации, материально-техническое обеспечение которой соответствует профилю программы.

Учебно-методическое и информационное обеспечение

По данному модулю используется электронный УМК. УМК предполагает использование разных типов материалов, сопровождающих учебный процесс, включая информационные, обучающие и контролирующие. На платформе электронных курсов размещаются задания, приводится перечень необходимых для изучения материалов. Обучающиеся могут на протяжении прохождения практики обращаться к теоретической базе знаний.

4. Оценка качества освоения программы практики (формы аттестации, оценочные и методические материалы)

В качестве подтверждения прохождения практики на базе предприятий, организаций, учреждений, для зачета результатов обучения предъявляется дневник прохождения практики (Приложение 2) (*отчет в виде дневника прохождения практики*).

Программу составили:

Программист отдела мобильной разработки
ООО Аспирити



З.Б. Новиков

Руководитель программы:
Программист отдела мобильной разработки
ООО Аспирити



З.Б. Новиков

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Наименование образовательной организации

Индивидуальный план слушателя, направляемого на практику

Фамилия, имя, отчество _____

Место работы и должность/статус _____

Название предприятия (организации), где проводится практика

Цель практики _____

Срок практики с «___» _____ 202 г. по «___» _____ 202 г.

План практики

№ п.п.	Перечень разрабатываемых (изучаемых) вопросов, виды работ	Количество часов	Форма отчета
1.			Дневник практики
2.			
3.			

СОГЛАСОВАНО

(должность ответственного)

(подпись)

*(расшифровка подписи) лица,
направляющего на практику)*

Наименование площадки

УТВЕРЖДАЮ
 Руководитель площадки
 _____ ФИО
 «_____» _____ 2022 г.
 М.П.

**ДНЕВНИК
 прохождения практики**

_____,
 (фамилия, имя, отчество)
 проходящего обучение в рамках дополнительной профессиональной программе
 переподготовки «Python в производстве: автоматизация и аналитика»

Цель практики:

Руководители практики (от организации): _____
 (должность) (ФИО)

1. Дневник

Дата	Выполняемая работа	Вопросы для консультантов и руководителей практики

2. Краткий отчет о практике

3. Заключение руководителя практики от принимающей организации

Руководитель практики

(подпись)

(расшифровка подписи)