МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ГЕОДЕЗИИ И КАРТОГРАФИИ
(МИИГАиК)

Факультет геоинформатики и информационной безопасности
Кафедра информатики и геоинформационных технологий

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 5BFEE1D286E9C95A0C0D2C611E434BD5
Владелец: Степанченко Алексей Леонидович
Действителен: с 23.09.2026 по 17.12.2027

«УТВЕРЖДАЮ»
Проректор
А. Л. Степанченко
15 мая 2026 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) «Б1.В.06 ОСНОВЫ РАЗРАБОТКИ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ»

Уровень высшего образования: бакалавриат

Направление подготовки: 09.03.03 Прикладная информатика

Направленность (профиль) подготовки: Разработка программного обеспечения для обработки и анализа пространственных данных

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2026

Срок получения образования: 4 года

Объем: в зачетных единицах: 4 з.е.
в академических часах: 144 ак.ч.

Москва, 2026

Разработчики:

Старший преподаватель кафедры Информатики и геоинформационных технологий Стрельцов А. С.

Фонд оценочных средств по дисциплине (модулю) составлен в соответствии с требованиями ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утвержденного приказом Минобрнауки от 19.09.2017 № 922

Согласование и утверждение

№	Подразделение	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	факультет Геоинформатики и информационной безопасности	Председатель ученого совета факультета	Матерухин А. В.	Рассмотрено	05.11.2025, № 12-25

1. Планируемые результаты обучения, соотнесенные с планируемыми результатами освоения образовательной программы

ПК-2 Способен разрабатывать требования и осуществлять проектирование программного обеспечения

ПК-2.1 Формализует и анализирует высокоуровневые требования к программному обеспечению

Знать:

ПК-2.1/Зн4 Современные способы визуализации пространственных данных и методы их реализации

Уметь:

ПК-2.1/Ум1 Планировать процессы и этапы обработки(получения) данных фотографической, оптико-электронной и других видов аэрокосмических съемок для решения определенного круга производственных и научных задач

Владеть:

ПК-2.1/Нв2 Навыками проведения анализа требований к системам распознавания образов

ПК-2.2 Обосновывает выбор системных решений и управляет архитектурными компромиссами при проектировании программного обеспечения

Знать:

ПК-2.2/Зн3 Базовые понятия компьютерной графики

Уметь:

ПК-2.2/Ум2 Обосновывать выбор архитектурных решений при построении систем распознавания образов

Владеть:

ПК-2.2/Нв1 Методами сбора и оценки качества пространственной информации, имеющей значения для подготовки и реализации проектов в соответствующих сферах профессиональной деятельности

ПК-2.3 Разрабатывает детальные функциональные и логические спецификации программных компонентов, включая описание алгоритмов, структур данных и интерфейсов взаимодействия

Знать:

ПК-2.3/Зн2 Методы описания алгоритмов и основные принципы построения функциональных и логических спецификаций программного обеспечения

Уметь:

ПК-2.3/Ум2 Математически грамотно формулировать задачу распознавания, формализовать алгоритмы распознавания в виде пошаговых описаний

Владеть:

ПК-2.3/Нв3 Использовать графические инструменты для построения изображения

2. Показатели и критерии оценивания компетенций

Планируемые результаты освоения программы	Уровень овладения	Критерии оценивания (перечень сформированных ИДК/ЗУВ по уровням освоения)

ПК-2 Способен разрабатывать требования и осуществлять проектирование программного обеспечения	Повышенный	ИДК ПК-2.1 Формализует и анализирует высокоуровневые требования к программному обеспечению Знать: ПК-2.1/Зн4 Современные способы визуализации пространственных данных и методы их реализации Уметь: ПК-2.1/Ум1 Планировать процессы и этапы обработки(получения) данных фотографической, оптико-электронной и других видов аэрокосмических съемок для решения определенного круга производственных и научных задач Владеть: ПК-2.1/Нв2 Навыками проведения анализа требований к системам распознавания образов ИДК ПК-2.2 Обосновывает выбор системных решений и управляет архитектурными компромиссами при проектировании программного обеспечения Знать: ПК-2.2/Зн3 Базовые понятия компьютерной графики Уметь: ПК-2.2/Ум2 Обосновывать выбор архитектурных решений при построении систем распознавания образов Владеть: ПК-2.2/Нв1 Методами сбора и оценки качества пространственной информации, имеющей значения для подготовки и реализации проектов в соответствующих сферах профессиональной деятельности ИДК ПК-2.3 Разрабатывает детальные функциональные и логические спецификации программных компонентов, включая описание алгоритмов, структур данных и интерфейсов взаимодействия Знать: ПК-2.3/Зн2 Методы описания алгоритмов и основные принципы построения функциональных и логических спецификаций программного обеспечения Уметь: ПК-2.3/Ум2 Математически грамотно формулировать задачу распознавания, формализовать алгоритмы распознавания в виде пошаговых описаний Владеть: ПК-2.3/Нв3 Использовать графические инструменты для построения изображения

	Базовый	ИДК ПК-2.1 Формализует и анализирует высокоуровневые требования к программному обеспечению Знать: ПК-2.1/Зн4 Современные способы визуализации пространственных данных и методы их реализации Уметь: ПК-2.1/Ум1 Планировать процессы и этапы обработки(получения) данных фотографической, оптико-электронной и других видов аэрокосмических съемок для решения определенного круга производственных и научных задач ИДК ПК-2.2 Обосновывает выбор системных решений и управляет архитектурными компромиссами при проектировании программного обеспечения Знать: ПК-2.2/Зн3 Базовые понятия компьютерной графики Уметь: ПК-2.2/Ум2 Обосновывать выбор архитектурных решений при построении систем распознавания образов ИДК ПК-2.3 Разрабатывает детальные функциональные и логические спецификации программных компонентов, включая описание алгоритмов, структур данных и интерфейсов взаимодействия Знать: ПК-2.3/Зн2 Методы описания алгоритмов и основные принципы построения функциональных и логических спецификаций программного обеспечения Уметь: ПК-2.3/Ум2 Математически грамотно формулировать задачу распознавания, формализовать алгоритмы распознавания в виде пошаговых описаний
	Пороговый	ИДК ПК-2.1 Формализует и анализирует высокоуровневые требования к программному обеспечению Знать: ПК-2.1/Зн4 Современные способы визуализации пространственных данных и методы их реализации ИДК ПК-2.2 Обосновывает выбор системных решений и управляет архитектурными компромиссами при проектировании программного обеспечения Знать: ПК-2.2/Зн3 Базовые понятия компьютерной графики

		ИДК ПК-2.3 Разрабатывает детальные функциональные и логические спецификации программных компонентов, включая описание алгоритмов, структур данных и интерфейсов взаимодействия Знать: ПК-2.3/Зн2 Методы описания алгоритмов и основные принципы построения функциональных и логических спецификаций программного обеспечения

3. Шкала оценивания

Оценка	Критерии оценивания (перечень сформированных ИДК/ЗУВ)
Зачтено	ПК-2.1 Формализует и анализирует высокоуровневые требования к программному обеспечению ПК-2.1/Зн4 Современные способы визуализации пространственных данных и методы их реализации ПК-2.2 Обосновывает выбор системных решений и управляет архитектурными компромиссами при проектировании программного обеспечения ПК-2.2/Зн3 Базовые понятия компьютерной графики ПК-2.3 Разрабатывает детальные функциональные и логические спецификации программных компонентов, включая описание алгоритмов, структур данных и интерфейсов взаимодействия ПК-2.3/Зн2 Методы описания алгоритмов и основные принципы построения функциональных и логических спецификаций программного обеспечения
Не зачтено	Результаты обучения не сформированы на достаточном уровне

4. Контрольные мероприятия по дисциплине

Вид контроля	Форма контроля/Оценочное средство
Текущий контроль	Выполнение контрольных нормативов Просмотр выполненных работ
Промежуточная аттестация	Зачет

№ п/п	Наименование раздела	Контролируемые ИДК	Вид контроля/ используемые оценочные материалы	
			Текущий	Промежут. аттестация
1	Введение в кроссплатформенную мобильную разработку	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет
2	Построение пользовательского интерфейса	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет
3	Навигация и управление состоянием	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет
4	Работа с данными	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет

5	Архитектурные паттерны	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет
6	Работа с устройством	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет
7	Интеграция методов машинного обучения	ПК-2.1 ПК-2.2 ПК-2.3	Выполнение контрольных нормативов Просмотр выполненных работ	Зачет
8	Тестирование и публикация	ПК-2.1 ПК-2.2 ПК-2.3	Просмотр выполненных работ	Зачет

5. Порядок проведения промежуточной аттестации

Порядок проведения промежуточной аттестации регламентируется локальными нормативными актами Университета:
- Положением о формах, периодичности, порядке текущего контроля успеваемости и промежуточной аттестации обучающихся по основным образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.6 «Организация и проведение промежуточной аттестации».
- Положением о порядке осуществления образовательного процесса, проведения текущего контроля успеваемости и промежуточной аттестации обучающихся с применением электронного обучения и дистанционных образовательных технологий в федеральном государственном бюджетном образовательном учреждении высшего образования «Московский государственный университет геодезии и картографии», п.8 «Проведение промежуточной аттестации с применением ЭО и ДОТ».

Промежуточная аттестация: Зачет, 8 семестр.
Зачёт по дисциплине проводится в формате тестирования для итоговой оценки знаний.

Студентам будет предложен тест, состоящий из 20 вопросов. На выполнение всей работы отводится шестьдесят минут, в течение которых запрещается использование любых вспомогательных материалов и электронных устройств.

Итоговый результат зачёта определяется в зависимости от процента правильных ответов. Для получения отметки «зачтено» необходимо верно ответить не менее чем на шестьдесят процентов вопросов, что соответствует 12 и более правильным ответам из 20. Результат ниже шестидесяти процентов правильных ответов, то есть 11 и менее верных ответов, признаётся неудовлетворительным и влечёт выставление отметки «не зачтено».

6. Оценочные материалы текущего контроля

Раздел 1. Введение в кроссплатформенную мобильную разработку
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 1.1. Обзор мобильных платформ и подходов к разработке
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 1.2. Установка и настройка среды разработки
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 1.3. Основы языка Dart
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 2. Построение пользовательского интерфейса
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 2.1. Виджеты Flutter
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 2.2. Компоновка интерфейса
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 2.3. Material Design и адаптивный интерфейс
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 2.4. Работа с пользовательским вводом
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 3. Навигация и управление состоянием
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 3.1. Навигация между экранами
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 3.2. Управление состоянием с Provider
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 3.3. Архитектура BLoC
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 4. Работа с данными
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 4.1. Локальное хранение данных
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 4.2. Сетевое взаимодействие
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 4.3. Интеграция с Supabase
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 5. Архитектурные паттерны
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 5.1. Паттерн MVC
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 5.2. Паттерн MVVM
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 5.3. Clean Architecture
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 6. Работа с устройством
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 6.1. Геолокация и карты
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 6.2. Камера и галерея
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 6.3. Сенсоры и уведомления
Форма контроля/оценочное средство: Просмотр выполненных работ

Раздел 7. Интеграция методов машинного обучения
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 7.1. TensorFlow Lite в Flutter
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 7.2. Обработка изображений с ML
Форма контроля/оценочное средство: Выполнение контрольных нормативов

Раздел 8. Тестирование и публикация
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3

Тема 8.1. Тестирование приложений
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 8.2. Отладка и профилирование
Форма контроля/оценочное средство: Просмотр выполненных работ

Тема 8.3. Сборка и публикация
Форма контроля/оценочное средство: Просмотр выполненных работ

7. Оценочные материалы промежуточной аттестации

8 семестр, Зачет
Контролируемые ИДК: ПК-2.1 ПК-2.2 ПК-2.3
Вопросы/Задания:

1. Какой язык программирования используется для разработки Flutter-приложений?

2. Как называется файл Flutter-проекта, содержащий список зависимостей (пакетов) проекта?

3. Как называется официальный набор инструментов для отладки и профилирования производительности Flutter-приложений, включающий анализ дерева виджетов и потребления памяти?

4. Дополните предложение термином:
Оператор _____ в Dart используется для безопасного доступа к значению

nullable-переменной, возвращая null вместо возбуждения исключения, если переменная не инициализирована.

5. Дополните предложение термином:

Метод жизненного цикла StatefulWidget _____ вызывается первым, сразу при создании состояния виджета, и обычно используется для однократной инициализации.

6. Раскройте содержание концепции дерева виджетов (widget tree) и объекта BuildContext в Flutter: дайте определение, поясните их взаимосвязь.

7. Опишите проблему управления состоянием в Flutter-приложениях и раскройте содержание паттерна Provider как одного из способов её решения: дайте определение ChangeNotifier и Consumer, поясните их взаимодействие.

8. Опишите архитектуру BLoC (Business Logic Component): раскройте содержание понятий Event и State и поясните, как они связаны между собой в этой архитектуре.

9. Дайте определение архитектурного паттерна Clean Architecture, опишите его три основных слоя (Domain, Data, Presentation) и поясните роль внедрения зависимостей (Dependency Injection) в обеспечении тестируемости такой архитектуры.

10. Дайте определение процесса сборки Flutter-приложения в форматы APK и App Bundle, поясните различие режимов Debug и Release, а также раскройте назначение профилирования производительности приложения.

11. Расположите этапы отображения нового кадра пользовательского интерфейса Flutter после изменения состояния StatefulWidget в правильной последовательности.

1. Перестроение (повторный вызов метода build()) затронутой части дерева виджетов.
2. Событие, инициирующее изменение данных (например, нажатие пользователем кнопки).
3. Отрисовка (рендеринг) обновлённого дерева виджетов на экране устройства.
4. Вызов метода setState() с изменением значений внутри объекта состояния.
5. Планирование Flutter-движком перестроения затронутого поддерева виджетов на следующий кадр.

12. Расположите этапы навигации на новый экран с последующим возвратом данных на предыдущий экран в правильной последовательности.

1. Возврат данных на предыдущий экран вызовом Navigator.pop() с передаваемым значением.
2. Переход на новый экран вызовом Navigator.push().
3. Ожидание результата вызовом await на стороне предыдущего экрана и получение переданных данных.

13. Расположите этапы получения данных с внешнего REST API и их отображения в интерфейсе Flutter-приложения в правильной последовательности.

1. Разбор (декодирование) полученного JSON-ответа в объекты Dart.
2. Отправка HTTP-запроса к серверу с использованием пакета http.
3. Обновление состояния приложения полученными данными и перестроение интерфейса.
4. Получение ответа сервера с данными в формате JSON.

14. Расположите этапы подключения проекта к облачной платформе Supabase и аутентификации пользователя в правильной последовательности.

1. Отправка данных для входа (например, email и пароль) в Supabase через вызов метода аутентификации.
2. Создание проекта в Supabase и получение ключей доступа (URL и API-ключа).
3. Проверка учётных данных на стороне Supabase и возврат токена сессии в случае успеха.
4. Инициализация клиента Supabase в Flutter-приложении с указанием полученных ключей.
5. Сохранение токена сессии на устройстве для последующих запросов от имени пользователя.

6. Отображение экрана авторизации (ввод email и пароля) пользователю приложения.

15. Расположите этапы получения фотографии с камеры устройства и её сохранения с использованием пакета image_picker в правильной последовательности.

1. Вызов метода pickImage() с указанием источника ImageSource.camera.
2. Запрос и получение разрешения на использование камеры устройства.
3. Получение объекта изображения (XFile) с путём к сделанному снимку.
4. Сохранение полученного изображения в постоянное хранилище приложения.

16. Установите соответствие между виджетом компоновки Flutter и его назначением.

Row - Заставляет дочерний виджет занять всё оставшееся доступное пространство внутри Row или Column

Column - Накладывает дочерние виджеты друг на друга (позиционирование поверх друг друга)

Stack - Располагает дочерние виджеты в один вертикальный столбец

Expanded - Располагает дочерние виджеты в один горизонтальный ряд

17. Установите соответствие между архитектурным паттерном и его ключевой особенностью.

MVC - ViewModel предоставляет представлению данные через механизм связывания (data binding), не будучи напрямую осведомлённым о конкретном представлении

MVVM - Приложение разделяется на независимые слои с направлением зависимостей от внешних слоёв к внутренним, обеспечивая независимость бизнес-логики от деталей реализации

Clean Architecture - Контроллер напрямую обрабатывает пользовательский ввод и обновляет модель, а представление напрямую обращается к модели для отображения данных

18. Установите соответствие между пакетом Flutter и его назначением.

Geolocator - Выполнение сетевых запросов к внешним серверам и API

image_picker - Получение текущих координат геолокации устройства

dio - Выбор изображений из галереи или съёмка фото с камеры устройства

shared_preferences - Загрузка и выполнение моделей машинного обучения в формате TensorFlow Lite

tflite_flutter - Хранение простых данных в формате «ключ – значение» на устройстве

19. Установите соответствие между видом тестирования Flutter-приложения (или проверки его работы) и его характеристикой.

Unit-тест - Измеряет расход вычислительных ресурсов (время построения кадра, потребление памяти) уже работающего приложения

Widget-тест - Проверяет работу приложения целиком в условиях, близких к реальным, на устройстве или эмуляторе

Integration-тест - Проверяет отдельную функцию или класс бизнес-логики в изоляции, без запуска интерфейса

Профилирование производительности - Проверяет отображение и поведение отдельного виджета в эмулированном дереве виджетов, без запуска на реальном устройстве

20. Установите соответствие между компонентом асинхронного программирования в Dart и его назначением

Future - Представляет последовательность асинхронных событий (значений), которые могут поступать многократно с течением времени

Stream - Приостанавливает выполнение асинхронной функции до завершения указанной операции, не блокируя при этом остальную часть приложения

await - Представляет результат асинхронной операции, который станет доступен один раз в будущем