



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования

«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

ФГБОУ ВО «ИГУ»

Кафедра естественнонаучных дисциплин

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

по дисциплине ФТД.02 Практические аспекты программирования

направление подготовки 09.03.03 Прикладная информатика

направленность (профиль)

Прикладная информатика (разработка программного обеспечения)

Иркутск – 2026

Одобрено УМК факультета бизнес-коммуникаций и информатики

Председатель УМК  М.Г. Синчурина

Разработан в соответствии с ФГОС ВО и с учетом требований профессионального стандарта 06.001 «Программист» и компетентностной модели ОПОП.

Разработчик:  Балахчи А.Г.

Цель фонда оценочных средств. Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу факультативной дисциплины «ФТД.02 Практические аспекты программирования». Перечень видов оценочных средств соответствует рабочей программе дисциплины.

Фонд оценочных средств включает материалы для текущего контроля в форме практических заданий, проверки репозитория, code review, заданий на тестирование и отладку, мини-проектов и заданий по применению ИИ, а также материалы для промежуточной аттестации в форме зачета в 1 и 2 семестрах.

1. Паспорт фонда оценочных средств

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-1 ППК-Р1 Способен разрабатывать и отлаживать программный код	ПК-1 ППК-Р1.1 Выполняет формализацию и алгоритмизацию поставленных задач для разработки программного кода	З-1. Знает алгоритмы решения типичных задач, области и способы их применения; З-2. знает нотации и ПО для графического отображения алгоритмов; З-3. знает методы и приемы алгоритмизации; У-1. умеет использовать методы формализации и алгоритмизации; У-2. умеет применять алгоритмы решения типовых задач.
	ПК-1 ППК-Р1.2 Формализует задачу ИТ-отрасли в язык естественнонаучных дисциплин	З-1. Знает основные разделы математики, применяемые для анализа и моделирования; У-1. умеет выбирать адекватный математический аппарат для формализованного описания; В-1. владеет навыками описания задач в виде формальных моделей, пригодных для алгоритмизации и программной реализации.
	ПК-1 ППК-Р1.3 Осуществляет обоснованный выбор методов и алгоритмов для программной реализации формальной математической модели	З-1. Знает основные классы методов программной реализации моделей и критерии выбора алгоритмов; У-1. умеет проводить сравнительный анализ и обоснование выбора алгоритмов; В-1. владеет методами адаптации методов и алгоритмов под специфику задачи; В-2. владеет навыками оценки эффективности выбранных алгоритмов.
	ПК-1 ППК-Р1.4 Разрабатывает программный код с использованием языков программирования	З-1. Знает синтаксис выбранного языка, особенности программирования и стандартные библиотеки; З-2. знает методологии разработки ПО; З-3. знает технологии программирования; У-1. умеет применять языки программирования для написания кода; У-2. умеет использовать среду программирования; У-3. умеет использовать возможности программно-технической архитектуры для написания кода.
	ПК-1 ППК-Р1.5 Оформляет программный код в соответствии с установленными требованиями	З-1. Знает нормативно-технические документы, определяющие требования к оформлению кода; З-2. знает стандарты оформления технической документации; У-1. умеет применять стандарты и шаблоны документации; У-2. умеет применять регламенты оформления кода; У-3. умеет применять инструментарий для создания и актуализации исходных текстов.
	ПК-1 ППК-Р1.6 Работает с системой управления версиями программного кода	З-1. Знает возможности системы управления версиями и вспомогательных средств; З-2. знает регламент использования системы управления версиями; У-1. умеет регистрировать изменения; У-2. умеет сохранять изменения в соответствии с регламентом; У-3. умеет выполнять слияние, разделение и сравнение исходных текстов.
	ПК-1 ППК-Р1.7 Проверяет и отлаживает программный код	З-1. Знает методы и приемы отладки; З-2. знает типы и форматы сообщений об ошибках и предупреждений; З-3. знает способы использования технологических журналов; У-1. умеет выявлять ошибки; У-2. умеет отлаживать код на уровне модулей; У-3. умеет отлаживать межмодульные взаимодействия и взаимодействия с окружением.
ПК-8 ППК-Р7 Способен применять искусственный интеллект	ПК-8 ППК-Р7.1 Применяет ИИ-инструменты для генерации программного кода	З-1. Знает принципы работы современных генеративных ИИ-моделей для генерации кода; З-2. знает ограничения и риски ИИ-генерации (безопасность, качество, лицензирование); У-1.

Компетенция	Индикаторы компетенций	Результаты обучения
(ИИ) для генерации и отладки программного кода		умеет формулировать корректные запросы для генерации кода; У-2. умеет интегрировать ИИ-инструменты в среду разработки.
	ПК-8 ППК-Р7.2 Использует ИИ для анализа и отладки кода	3-1. Знает методы ИИ-анализа кода; 3-2. знает форматы и инструменты автоматизированного тестирования с ИИ; У-1. умеет настраивать ИИ-инструменты для поиска уязвимостей; У-2. умеет интерпретировать рекомендации ИИ по исправлению кода.
	ПК-8 ППК-Р7.3 Оптимизирует код с помощью ИИ	3-1. Знает методы ИИ-оптимизации; 3-2. знает критерии качества кода, применяемые ИИ-системами; У-1. умеет использовать ИИ для рефакторинга; У-2. умеет проверять корректность оптимизаций, предложенных ИИ.
	ПК-8 ППК-Р7.4 Оценивает этические и профессиональные аспекты применения ИИ в разработке	3-1. Знает этические нормы использования ИИ; 3-2. знает лицензионные ограничения сгенерированного кода; У-1. умеет проверять код на соответствие стандартам после ИИ-генерации; У-2. умеет документировать использование ИИ в разработке.

Измеримые результаты освоения дисциплины

Код	Результат обучения	Индикаторы
РО-1	Формализует прикладную задачу, декомпозирует ее и разрабатывает алгоритм решения, выбирая подходящие структуры данных и оценивая ограничения решения.	ПК-1 ППК-Р1.1; ПК-1 ППК-Р1.2; ПК-1 ППК-Р1.3
РО-2	Разрабатывает работоспособный программный код с использованием выбранного языка, стандартных библиотек и средств среды разработки.	ПК-1 ППК-Р1.4
РО-3	Оформляет исходный код и техническое описание решения, ведет историю изменений в Git и использует ветвление/слияние при выполнении практических задач.	ПК-1 ППК-Р1.5; ПК-1 ППК-Р1.6
РО-4	Проверяет, тестирует и отлаживает программный код, интерпретирует сообщения об ошибках и данные журналирования, исправляет выявленные дефекты.	ПК-1 ППК-Р1.7
РО-5	Применяет ИИ-инструменты для генерации фрагментов кода, тестов и вариантов решения, формулирует запросы и проверяет корректность полученного результата.	ПК-8 ППК-Р7.1; ПК-8 ППК-Р7.2
РО-6	Использует ИИ для анализа и рефакторинга кода, сравнивает исходное и предложенное решения, проверяет безопасность, качество и лицензионно-этические ограничения и документирует использование ИИ.	ПК-8 ППК-Р7.2; ПК-8 ППК-Р7.3; ПК-8 ППК-Р7.4

2. Показатели и критерии оценивания компетенций, шкалы оценивания

2.1. Показатели и критерии оценивания компетенций

№	Раздел, тема	Код индикатора компетенции	Текущий контроль	Промежуточная аттестация
1	Среда разработки и рабочий процесс программиста	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—
2	Базовые конструкции языка и решение типовых задач	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—
3	Функции, модули и структуры данных	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—
4	Алгоритмизация прикладных задач	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—
5	Тестирование, отладка и качество кода	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—
6	Практический спринт I: мини-проект	ПК-1 ППК-Р1.1–Р1.7; ПК-8 ППК-Р7.1–Р7.4	Проект, защита	Зачет
7	Организация программного кода и объектная декомпозиция	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—
8	Работа с файлами, форматами данных и внешними ресурсами	ПК-1 ППК-Р1.1–Р1.7	Практическое задание	—

№	Раздел, тема	Код индикатора компетенции	Текущий контроль	Промежуточная аттестация
9	Обработка данных и автоматизация прикладной задачи	ПК-1 ППК-P1.1–P1.7	Практическое задание	—
10	Командная разработка: Git, ветвление, слияние, review	ПК-1 ППК-P1.5; ПК-1 ППК-P1.6; ПК-1 ППК-P1.7	Репозиторий, code review	—
11	ИИ в практике программирования	ПК-1 ППК-P1.1–P1.7; ПК-8 ППК-P7.1–P7.4	Практическое задание, журнал ИИ	—
12	Практический спринт II: интегрированный проект	ПК-1 ППК-P1.1–P1.7; ПК-8 ППК-P7.1–P7.4	Проект, защита	Зачет

2.2. Критерии оценивания результатов обучения для текущего контроля успеваемости и промежуточной аттестации

Оценочное средство	Критерии оценивания	Шкала оценивания
Практическое задание	Корректность результата — 40%; полнота реализации требований — 20%; качество и читаемость кода — 15%; тестирование и обработка граничных случаев — 15%; объяснение решения — 10%.	5: 90–100%; 4: 80–89%; 3: 70–79%; 2: менее 70% либо решение неработоспособно/не объяснено.
Репозиторий / Git	Осмысленные коммиты; корректная структура веток; отсутствие случайных/служебных файлов; возможность восстановить ход работы; корректное слияние; оформление README.	Зачтено: выполнено не менее 70% требований и студент объясняет историю изменений. Не зачтено: история не подтверждает выполнение работы или студент не может объяснить ключевые изменения.
Code review	Выявление ошибок и рисков; конкретность замечаний; приоритет замечаний; корректность предлагаемых исправлений; профессиональная форма обратной связи.	5: замечания полные и обоснованные; 4: есть отдельные пропуски; 3: выявлены основные проблемы; 2: замечания поверхностные или ошибочные.
Задание с ИИ	Качество постановки запроса — 15%; проверка и тестирование результата — 30%; самостоятельная корректировка — 20%; анализ рисков/ограничений — 15%; фиксация использования ИИ — 10%; способность объяснить код — 10%.	Зачтено: не менее 70% и отсутствует критический дефект, скрытый за ИИ-генерацией. Не зачтено: студент не понимает предложенный код, не проверил его или нарушил требования к данным/лицензированию.
Проект	Соответствие требованиям — 25%; корректность и устойчивость — 20%; алгоритм/структура решения — 15%; качество кода — 10%; тесты и отладка — 10%; Git и воспроизводимость — 5%; документация — 5%; обоснование решений и защита — 10%.	Зачтено: итоговый балл не менее 70%, выполнены критические требования и студент подтверждает авторство/понимание решения. Не зачтено: менее 70%, критические функции не работают либо защита не подтверждает освоение результатов.

2.3. Оценочные средства для текущего контроля (примеры)

2.3.1. Практические задания базового уровня

Задание Б-1. По заданному набору входных данных реализовать вычисление результата с использованием условий и циклов. Подготовить не менее пяти контрольных примеров, включая граничные случаи.

Задание Б-2. Реализовать функцию обработки строки согласно спецификации. Добавить модульные тесты для обычных и ошибочных входных данных.

Задание Б-3. Получить список числовых значений, удалить некорректные элементы, вычислить агрегаты и сформировать текстовый отчет.

Задание Б-4. Разбить монолитный фрагмент программы на функции. Обосновать границы функций и имена параметров.

Задание Б-5. Получить программу с синтаксическими и логическими дефектами. Воспроизвести ошибки, локализовать причины и исправить код, сохранив отдельные коммиты на каждую группу исправлений.

Задание Б-6. Создать Git-репозиторий, выполнить не менее пяти логически завершенных коммитов и показать различия между двумя версиями программы.

2.3.2. Практические задания прикладного уровня

Задание П-1. Валидатор данных

Разработать программу, читающую CSV/JSON-файл, проверяющую обязательные поля и типы значений, формирующую отчет об ошибках и итоговый корректный набор данных. Самостоятельно выбрать структуру данных и способ обработки ошибок.

Задание П-2. Анализ журнала

Разработать анализатор текстового журнала: выделить тип события, время и идентификатор; подсчитать статистику; вывести топ повторяющихся ошибок. Предусмотреть поврежденные строки.

Задание П-3. Рефакторинг

Дан работающий, но плохо структурированный программный модуль. Сохранить поведение, выделить функции/классы, улучшить имена, устранить дублирование, добавить тесты до и после изменений.

Задание П-4. Code review

Получить merge request одногруппника или учебный пример. Провести review по чек-листу: корректность, читаемость, обработка ошибок, тестируемость, безопасность ввода. Отметить минимум пять содержательных замечаний и предложить исправления.

Задание П-5. Слияние изменений

В двух ветках изменена одна функциональность разными способами. Выполнить анализ, разрешить конфликт с сохранением корректного поведения, запустить тесты и оформить сообщение о слиянии.

Задание П-6. ИИ: генерация + проверка

Сформулировать промпт для генерации функции и тестов. Сохранить исходный запрос и ответ. Проверить код на скрытом наборе случаев, выявить не менее одного ограничения или риска, доработать решение самостоятельно.

Задание П-7. ИИ: поиск дефекта

Передать ИИ фрагмент кода с несколькими дефектами. Сравнить рекомендации ИИ с результатами собственного анализа и отладчика. Принять только подтвержденные исправления и зафиксировать основания.

Задание П-8. ИИ: рефакторинг

Получить от ИИ два варианта рефакторинга. Сравнить читаемость, сложность, количество тестов, риски изменения поведения. Выбрать или отклонить варианты и аргументировать решение.

2.3.3. Профессионально-ориентированные задания

ПРОФ-1. Диагностика инцидента

В учебном приложении периодически возникает ошибка при обработке входных данных. Доступны код, набор журналов и описание наблюдаемого поведения. Требуется воспроизвести дефект, построить гипотезы, локализовать причину, внести минимальное исправление, добавить регрессионный тест и оформить изменения в отдельной ветке.

ПРОФ-2. Улучшение качества кода

Получен модуль, прошедший функциональные тесты, но имеющий дублирование, длинные функции и слабую обработку ошибок. Требуется выполнить code review, составить план рефакторинга, провести изменения маленькими коммитами, доказать сохранение поведения тестами и оформить итоговое заключение.

ПРОФ-3. ИИ-ассистированная разработка

Требуется реализовать новую функцию в существующем учебном проекте. Студент самостоятельно определяет, какие этапы целесообразно выполнить с ИИ (варианты архитектуры, код, тесты, документация), фиксирует запросы и ответы, проверяет каждый принятый фрагмент и в защите объясняет, какие рекомендации отклонены и почему.

2.3.4. Чек-лист проверки репозитория

- репозиторий содержит понятную структуру проекта и README с инструкцией запуска;
- секреты, токены, персональные данные и временные файлы не включены в историю;
- коммиты отражают этапы работы и имеют осмысленные сообщения;
- для существенных изменений использованы отдельные ветки;
- конфликты слияния разрешены без потери требуемой функциональности;
- тесты запускаются воспроизводимо;
- при использовании ИИ имеется журнал существенных запросов и принятых решений.

2.4. Оценочные средства для промежуточной аттестации

2.4.1. Зачет за 1 семестр

Форма: защита индивидуального мини-проекта и короткое собеседование по исходному коду. Проект должен быть выполнен в Git-репозитории и включать алгоритм/описание решения, исходный код, тесты или контрольные примеры и README.

Типовое задание: разработать консольное приложение для учета и обработки набора сущностей (например, задачи, книги, заявки, события) с загрузкой исходных данных, поиском/фильтрацией, вычислением сводных показателей и сохранением результата. Конкретная предметная область выдается преподавателем. Обязательны обработка ошибочных входных данных и не менее 8 тестовых сценариев.

Условия зачета: выполнено не менее 70% обязательных текущих работ; проект работоспособен; критические требования выполнены; студент объясняет алгоритм и ключевой код; история Git подтверждает последовательную работу. Использование ИИ допускается только с фиксацией и объяснением принятых фрагментов.

2.4.2. Зачет за 2 семестр

Форма: защита интегрированного проекта, демонстрация репозитория и техническое собеседование. Задание имеет открытый характер: студент выбирает структуру решения и обосновывает ключевые технические решения.

Типовое задание: разработать небольшое программное средство автоматизации прикладной задачи, которое получает данные из файла или внешнего источника, выполняет валидацию и преобразование, реализует не менее трех пользовательских операций и формирует результат/отчет. Код должен быть разделен на компоненты, покрыт тестами ключевой логики и подготовлен для совместной разработки через ветки и review.

Обязательная часть по ИИ: применить ИИ минимум на двух разных этапах (например, генерация тестов и анализ дефекта; рефакторинг и документация). Для каждого случая представить: исходный запрос; краткое содержание ответа; способ проверки; принятые изменения; отклоненные рекомендации; оценку рисков по качеству, безопасности, конфиденциальности и лицензированию.

Условия зачета: итоговая оценка проекта по рубрике не ниже 70%; нет критических функциональных дефектов; студент способен внести небольшое изменение в код во время защиты или объяснить способ его реализации; может аргументированно отличить собственные решения от предложений ИИ и показать, как они проверялись.

2.4.3. Примерные вопросы технического собеседования

1. Как вы декомпозировали задачу и почему выбрали именно такую структуру программы?
2. Какие граничные случаи вы учитывали и какими тестами они проверяются?
3. Покажите один дефект, который возникал в ходе разработки, и объясните процесс его локализации.
4. Почему выбранная структура данных подходит для этой задачи? Какие есть альтернативы?
5. Как по истории Git восстановить основные этапы вашей работы?
6. Какое изменение было самым рискованным и как вы убедились, что оно не сломало прежнее поведение?
7. Какие рекомендации ИИ вы отклонили? Почему?
8. Как вы проверяли код, полученный от ИИ, на корректность и безопасность?
9. Какие данные нельзя передавать во внешние ИИ-сервисы в реальном проекте?
10. Какие части решения вы бы улучшили при наличии дополнительного времени?

2.4.4. Матрица «результат обучения — оценочное задание — критерий»

Результат	Индикаторы	Оценочное задание	Ключевой критерий
PO-1	ПК-1 ППК-P1.1; ПК-1 ППК-P1.2; ПК-1 ППК-P1.3	Алгоритмические задачи; проект	корректность формализации, выбор и обоснование алгоритма, граничные случаи
PO-2	ПК-1 ППК-P1.4	Практические работы; проект	работоспособность, полнота требований, качество кода
PO-3	ПК-1 ППК-P1.5; ПК-1 ППК-P1.6	Репозиторий; code review; проект	оформление, история изменений, ветвление/слияние, документация
PO-4	ПК-1 ППК-P1.7	Задания на дефекты; тестирование; проект	воспроизведение, локализация, тест, корректность исправления
PO-5	ПК-8 ППК-P7.1; ПК-8 ППК-P7.2	ИИ: генерация + проверка	качество запроса, верификация, самостоятельная корректировка, объяснение
PO-6	ПК-8 ППК-P7.2; ПК-8 ППК-P7.3; ПК-8 ППК-P7.4	ИИ: рефакторинг; интегрированный проект	сравнение вариантов, проверка качества, безопасность/этика, журнал использования ИИ