



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ФГБОУ ВО «ИГУ»

Кафедра естественнонаучных дисциплин

УТВЕРЖДАЮ

Декан факультета бизнес-коммуникаций и
информатики

 М.Г. Синчурина

«18» марта 2026 г.

Рабочая программа дисциплины (модуля)

Наименование дисциплины (модуля)

**ФТД.02 Практические аспекты
программирования**

(индекс дисциплины по учебному плану, наименование дисциплины (модуля)).

Направление подготовки:

09.03.03 Прикладная информатика

(код, наименование направления подготовки)

Направленность (профиль) подготовки:

**Прикладная информатика (Разработка
программного обеспечения)**

Квалификация выпускника –бакалавр

Форма обучения: очная

(очная, заочная (с использованием электронного обучения и дистанционных образовательных технологий), очно-заочная (с использованием электронного обучения и дистанционных образовательных технологий *))*

Согласовано с УМК факультета
бизнес-коммуникаций и информатики

Рекомендовано кафедрой
естественнонаучных дисциплин:

Протокол № 7 от «18» марта 2026 г.

Протокол № 2 от «2» марта 2026 г.

Председатель  М.Г. Синчурина

и.о.зав. кафедрой  А.Г. Балахчи

СОДЕРЖАНИЕ

I. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ (МОДУЛЯ)

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

III. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

4.1 Содержание дисциплины, структурированное по темам

4.2 План внеаудиторной самостоятельной работы

4.3 Содержание учебного материала

4.3.1 Перечень практических занятий

4.3.2 Перечень тем, выносимых на самостоятельное изучение

4.4 Методические указания по организации самостоятельной работы

4.5 Примерная тематика курсовых работ (проектов)

V. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

VI. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

VII. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

VIII. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

I. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ (МОДУЛЯ)

Цель дисциплины — сформировать у обучающихся устойчивые практические навыки решения программных задач и организации личного рабочего процесса разработчика: от формализации задачи и выбора алгоритма до реализации, тестирования, отладки, ведения исходного кода в системе контроля версий и осознанного применения инструментов искусственного интеллекта.

Дисциплина имеет практико-ориентированный характер и дополняет базовый курс «Программирование» за счет большого объема практики, повторяющихся циклов «задача → алгоритм → код → тест → отладка → улучшение → фиксация результата» и формирования инженерных привычек, необходимых для дальнейшего освоения дисциплин профиля разработки программного обеспечения.

- закрепить навыки формализации и алгоритмизации задач, разработки программного кода и выбора способа решения;
- сформировать практику регулярного использования среды разработки, отладчика, линтеров, тестов и журналирования;
- освоить рабочий цикл Git: фиксация изменений, ветвление, сравнение, слияние и подготовка изменений к проверке;
- развить навыки декомпозиции программ, структурирования исходного кода, документирования и code review;
- научить применять ИИ-инструменты как вспомогательное средство для генерации, анализа, отладки и рефакторинга кода с обязательной верификацией результата;
- сформировать ответственное отношение к конфиденциальности, лицензированию, качеству и воспроизводимости решений при использовании ИИ;
- обеспечить накопление портфолио практических работ и двух завершенных программных проектов в репозитории.

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Факультативная дисциплина «ФТД.02 Практические аспекты программирования» относится к факультативным дисциплинам ОПОП по направлению 09.03.03 «Прикладная информатика», профиль «Прикладная информатика (разработка программного обеспечения)». Дисциплина реализуется в 1 и 2 семестрах и ориентирована на профессиональную роль инженер-программиста. В компетентностной модели эта роль связана с профессиональным стандартом 06.001 «Программист», прежде всего с трудовыми действиями по формализации, написанию, оформлению, версионированию, проверке и отладке программного кода.

Практическая направленность дисциплины обеспечивает раннее включение студента в типовой рабочий процесс разработчика. Задания выполняются в виде коротких программных задач, кейсов, code review и мини-проектов. Результаты фиксируются в репозитории, а применение ИИ рассматривается как контролируемый инженерный инструмент, а не как замена самостоятельной разработки.

2.2. Предшествующие дисциплины: специальных вузовских дисциплин не требуется, поскольку изучение начинается в 1 семестре. Используются базовые общеобразовательные навыки работы с компьютером, математические представления и элементы алгоритмического мышления.

2.3. Результаты дисциплины используются далее при освоении дисциплин профессионального цикла, в том числе «Объектно-ориентированный анализ и программирование», «Веб-программирование на стороне клиента», «Веб-программирование», «Базы данных (продвинутый уровень)», «Шаблоны проектирования», «Проектирование информационных систем»,

«Автоматизация разработки и тестирования с помощью ИИ», а также при выполнении проектных заданий и практик.

2.4. Смежные дисциплины, изучаемые параллельно: в 1 семестре — «Программирование»; во 2 семестре — дисциплины, использующие программирование как инструмент решения прикладных задач. Факультатив не дублирует теоретическое содержание основной дисциплины, а обеспечивает дополнительную практику разработки, тестирования, отладки и работы с инструментами разработчика.

III. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование элементов профессиональных компетенций, закрепленных за дисциплиной учебным планом: ПК-1 ППК-Р1 и ПК-8 ППК-Р7.

Компетенция	Индикатор компетенции	Результаты обучения, закрепленные в компетентностной модели
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	ПК-8 ППК-Р7.1	З-1. Знает принципы работы современных генеративных ИИ-

Компетенция	Индикатор компетенции	Результаты обучения, закрепленные в компетентностной модели
Способен применять искусственный интеллект (ИИ) для генерации и отладки программного кода	Применяет ИИ-инструменты для генерации программного кода	моделей для генерации кода; 3-2. знает ограничения и риски ИИ-генерации (безопасность, качество, лицензирование); У-1. 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.

Интегрированные измеримые результаты освоения дисциплины:

Код	Результат обучения	Связанные 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

IV. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Трудоемкость дисциплины составляет 6 зачетных единиц, 216 часов. Дисциплина реализуется в 1 и 2 семестрах: по 3 зачетные единицы (108 часов) в каждом семестре. В 1 семестре предусмотрено 90 часов практических занятий, 10 часов самостоятельной работы и 8 часов контроля; во 2 семестре — 84 часа практических занятий, 16 часов самостоятельной работы и 8 часов контроля. Форма промежуточной аттестации — зачет в 1 и 2 семестрах.

4.1 Содержание дисциплины, структурированное по темам, с указанием видов учебных занятий и СРС, отведенного на них количества академических часов

№	Раздел / тема	Семес-тр	Лекции	Практические занятия	Консультации	СРС	Краткое содержание	Форма контроля
1	Среда разработки и рабочий процесс программиста	1	0	10	0	1	Настройка среды; запуск, ввод/вывод, командная строка; структура проекта; базовая работа с Git.	Практическая работа, репозиторий

№	Раздел / тема	Семес тр	Лекци и	Практиче ские занятия	Консуль тации	СРС	Краткое содержание	Форма контроля
2	Базовые конструкции языка и решение типовых задач	1	0	16	0	2	Типы данных, выражения, условия, циклы. Разработка решений по спецификации и набору тестовых примеров.	Практическая работа, автоматическая проверка
3	Функции, модули и структуры данных	1	0	18	0	2	Функции, область видимости, коллекции, строки, декомпозиция, повторное использование кода.	Практическая работа, code review
4	Алгоритмизация прикладных задач	1	0	18	0	2	Декомпозиция, выбор алгоритма, оценка корректности и простая оценка эффективности; обработка граничных случаев.	Практическая работа, разбор решения
5	Тестирование, отладка и качество кода	1	0	16	0	1	Отладчик, сообщения об ошибках, assertions, модульные тесты, логирование, линтеры, правила оформления кода.	Практическая работа, тестирование кода
6	Практический спринт I: мини-проект	1	0	12	0	2	Разработка небольшого программного продукта по набору требований: алгоритм, реализация, тесты, Git, документация; первичное использование ИИ как помощника с обязательной проверкой результата.	Мини-проект, защита
7	Организация программного кода и объектная декомпозиция	2	0	16	0	2	Классы и объекты в прикладных задачах, интерфейсы компонентов, композиция, разделение ответственности, рефакторинг.	Практическая работа, code review
8	Работа с файлами, форматами данных и внешними ресурсами	2	0	14	0	2	Текстовые файлы, CSV/JSON, сериализация, обработка ошибок, получение данных из внешнего источника/API при наличии технической возможности.	Практическая работа
9	Обработка данных и автоматизация прикладной задачи	2	0	12	0	2	Построение программного конвейера: чтение, преобразование, проверка, агрегирование, формирование результата.	Практическая работа, кейс
10	Командная разработка: Git, ветвление, слияние, review	2	0	14	0	2	Feature-ветки, merge/pull request, разрешение конфликтов, требования к коммитам, инспекция кода, фиксация замечаний.	Репозиторий, code review
11	ИИ в практике программирования	2	0	14	0	4	Генерация и объяснение кода, тестов и документации; поиск дефектов; рефакторинг; проверка предложений ИИ; безопасность, конфиденциальность, лицензирование и фиксация использования ИИ.	Практическая работа, журнал применения ИИ
12	Практический спринт II: интегрированный проект	2	0	14	0	4	Разработка программного решения по открытой задаче с самостоятельным выбором алгоритмов и структуры, тестированием, Git, code review и контролируемым применением ИИ.	Проект, защита, собеседование
	Итого за 1 семестр	1	0	90	0	10	Контроль — 8 часов	Зачет
	Итого за 2 семестр	2	0	84	0	16	Контроль — 8 часов	Зачет
	ИТОГО		0	174	0	26	216 часов, включая 16 часов контроля	

4.2 План внеаудиторной самостоятельной работы обучающихся по дисциплине

Семестр	Тема	Вид самостоятельной работы	Часы	Оценочное средство
1	Среда разработки и рабочий процесс программиста	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	1	Практическая работа, репозиторий
1	Базовые конструкции языка и решение типовых задач	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Практическая работа, автоматическая проверка

Семестр	Тема	Вид самостоятельной работы	Часы	Оценочное средство
1	Функции, модули и структуры данных	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Практическая работа, code review
1	Алгоритмизация прикладных задач	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Практическая работа, разбор решения
1	Тестирование, отладка и качество кода	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	1	Практическая работа, тестирование кода
1	Практический спринт I: мини-проект	Проектирование, доработка и документирование мини-проекта; подготовка тестов, README и материалов к защите.	2	Мини-проект, защита
2	Организация программного кода и объектная декомпозиция	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Практическая работа, code review
2	Работа с файлами, форматами данных и внешними ресурсами	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Практическая работа
2	Обработка данных и автоматизация прикладной задачи	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Практическая работа, кейс
2	Командная разработка: Git, ветвление, слияние, review	Подготовка и завершение практической работы; анализ ошибок; работа с документацией; оформление репозитория.	2	Репозиторий, code review
2	ИИ в практике программирования	Ведение журнала использования ИИ: запрос, ответ, проверка, принятые/отклоненные изменения, обоснование.	4	Практическая работа, журнал применения ИИ
2	Практический спринт II: интегрированный проект	Проектирование, доработка и документирование мини-проекта; подготовка тестов, README и материалов к защите.	4	Проект, защита, собеседование

4.3 Содержание учебного материала

1. Среда разработки и рабочий процесс программиста

Настройка среды; запуск, ввод/вывод, командная строка; структура проекта; базовая работа с Git.

2. Базовые конструкции языка и решение типовых задач

Типы данных, выражения, условия, циклы. Разработка решений по спецификации и набору тестовых примеров.

3. Функции, модули и структуры данных

Функции, область видимости, коллекции, строки, декомпозиция, повторное использование кода.

4. Алгоритмизация прикладных задач

Декомпозиция, выбор алгоритма, оценка корректности и простая оценка эффективности; обработка граничных случаев.

5. Тестирование, отладка и качество кода

Отладчик, сообщения об ошибках, assertions, модульные тесты, логирование, линтеры, правила оформления кода.

6. Практический спринт I: мини-проект

Разработка небольшого программного продукта по набору требований: алгоритм, реализация, тесты, Git, документация; первичное использование ИИ как помощника с обязательной проверкой результата.

7. Организация программного кода и объектная декомпозиция

Классы и объекты в прикладных задачах, интерфейсы компонентов, композиция, разделение ответственности, рефакторинг.

8. Работа с файлами, форматами данных и внешними ресурсами

Текстовые файлы, CSV/JSON, сериализация, обработка ошибок, получение данных из внешнего источника/API при наличии технической возможности.

9. Обработка данных и автоматизация прикладной задачи

Построение программного конвейера: чтение, преобразование, проверка, агрегирование, формирование результата.

10. Командная разработка: Git, ветвление, слияние, review

Feature-ветки, merge/pull request, разрешение конфликтов, требования к коммитам, инспекция кода, фиксация замечаний.

11. ИИ в практике программирования

Генерация и объяснение кода, тестов и документации; поиск дефектов; рефакторинг; проверка предложений ИИ; безопасность, конфиденциальность, лицензирование и фиксация использования ИИ.

12. Практический спринт II: интегрированный проект

Разработка программного решения по открытой задаче с самостоятельным выбором алгоритмов и структуры, тестированием, Git, code review и контролируемым применением ИИ.

4.3.1. Перечень семинарских, практических занятий и лабораторных работ

Семестр	№	Практическая работа	Часы	Формируемые индикаторы
1	1.1	Настройка рабочего окружения и запуск проекта	4	ПК-1 / ПК-8 по содержанию работы
1	1.2	Командная строка и файловая структура проекта	3	ПК-1 / ПК-8 по содержанию работы
1	1.3	Создание Git-репозитория, коммиты, история изменений	3	ПК-1 / ПК-8 по содержанию работы
1	2.1	Ветвления и обработка входных данных	4	ПК-1 / ПК-8 по содержанию работы
1	2.2	Циклы и накопительные алгоритмы	4	ПК-1 / ПК-8 по содержанию работы
1	2.3	Практикум по строкам и числовым данным	4	ПК-1 / ПК-8 по содержанию работы
1	2.4	Набор задач с автоматической проверкой	4	ПК-1 / ПК-8 по содержанию работы
1	3.1	Функции и декомпозиция решения	5	ПК-1 / ПК-8 по содержанию работы
1	3.2	Коллекции и выбор структуры данных	5	ПК-1 / ПК-8 по содержанию работы
1	3.3	Модули и повторное использование кода	4	ПК-1 / ПК-8 по содержанию работы
1	3.4	Разбор и улучшение чужого решения	4	ПК-1 / ПК-8 по содержанию работы

Семестр	№	Практическая работа	Часы	Формируемые индикаторы
1	4.1	Формализация условия и построение алгоритма	5	ПК-1 / ПК-8 по содержанию работы
1	4.2	Граничные случаи и контрольные примеры	5	ПК-1 / ПК-8 по содержанию работы
1	4.3	Сравнение нескольких алгоритмов	4	ПК-1 / ПК-8 по содержанию работы
1	4.4	Практический кейс с самостоятельным выбором решения	4	ПК-1 / ПК-8 по содержанию работы
1	5.1	Работа с отладчиком	4	ПК-1 / ПК-8 по содержанию работы
1	5.2	Модульные тесты и assertions	4	ПК-1 / ПК-8 по содержанию работы
1	5.3	Логирование и диагностика	4	ПК-1 / ПК-8 по содержанию работы
1	5.4	Линтер, форматирование и исправление code smells	4	ПК-1 / ПК-8 по содержанию работы
1	6.1	Проектирование мини-проекта	4	ПК-1 / ПК-8 по содержанию работы
1	6.2	Реализация функциональности и тестов	4	ПК-1 / ПК-8 по содержанию работы
1	6.3	Подготовка репозитория и защита	4	ПК-1 / ПК-8 по содержанию работы
2	7.1	Объектная декомпозиция задачи	4	ПК-1 / ПК-8 по содержанию работы
2	7.2	Классы, композиция и интерфейсы	4	ПК-1 / ПК-8 по содержанию работы
2	7.3	Рефакторинг процедурного решения	4	ПК-1 / ПК-8 по содержанию работы
2	7.4	Code review архитектуры небольшого приложения	4	ПК-1 / ПК-8 по содержанию работы
2	8.1	Чтение и запись файлов	4	ПК-1 / ПК-8 по содержанию работы
2	8.2	CSV/JSON и проверка данных	4	ПК-1 / ПК-8 по содержанию работы
2	8.3	Обработка исключений	3	ПК-1 / ПК-8 по содержанию работы
2	8.4	Получение и обработка данных из внешнего источника/API	3	ПК-1 / ПК-8 по содержанию работы
2	9.1	Конвейер обработки данных	3	ПК-1 / ПК-8 по содержанию работы
2	9.2	Валидация и преобразование	3	ПК-1 / ПК-8 по содержанию работы
2	9.3	Агрегирование и формирование отчета	3	ПК-1 / ПК-8 по содержанию работы
2	9.4	Кейс автоматизации прикладной задачи	3	ПК-1 / ПК-8 по содержанию работы
2	10.1	Ветвление и стратегия работы с изменениями	4	ПК-1 / ПК-8 по содержанию работы
2	10.2	Merge/pull request и разрешение конфликтов	4	ПК-1 / ПК-8 по содержанию работы

Семестр	№	Практическая работа	Часы	Формируемые индикаторы
2	10.3	Code review по чек-листу	3	ПК-1 / ПК-8 по содержанию работы
2	10.4	Исправление замечаний и подготовка версии	3	ПК-1 / ПК-8 по содержанию работы
2	11.1	Промпты для генерации кода и тестов	4	ПК-1 / ПК-8 по содержанию работы
2	11.2	ИИ-анализ дефектов и объяснение ошибок	4	ПК-1 / ПК-8 по содержанию работы
2	11.3	ИИ-рефакторинг с верификацией результата	3	ПК-1 / ПК-8 по содержанию работы
2	11.4	Безопасность, конфиденциальность, лицензирование; журнал использования ИИ	3	ПК-1 / ПК-8 по содержанию работы
2	12.1	Анализ требований и выбор решения	4	ПК-1 / ПК-8 по содержанию работы
2	12.2	Реализация инкремента проекта	4	ПК-1 / ПК-8 по содержанию работы
2	12.3	Тестирование, отладка и code review	3	ПК-1 / ПК-8 по содержанию работы
2	12.4	Подготовка итоговой версии, документации и защиты	3	ПК-1 / ПК-8 по содержанию работы

4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение самостоятельной работы студентов

Семестр	Тема	Вопросы и виды работы	Часы
1	Среда разработки и рабочий процесс программиста	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	1
1	Базовые конструкции языка и решение типовых задач	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
1	Функции, модули и структуры данных	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
1	Алгоритмизация прикладных задач	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
1	Тестирование, отладка и качество кода	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	1
1	Практический спринт I: мини-проект	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
2	Организация программного кода и объектная декомпозиция	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
2	Работа с файлами, форматами данных и внешними ресурсами	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
2	Обработка данных и автоматизация прикладной задачи	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
2	Командная разработка: Git, ветвление, слияние, review	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	2
2	ИИ в практике программирования	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	4
2	Практический спринт II: интегрированный проект	Документация языка и используемых библиотек; разбор типовых ошибок; дополнительные задачи; оформление результатов практической работы.	4

4.4. Методические указания по организации самостоятельной работы студентов

Основная единица работы студента — завершенная программная задача или инкремент проекта, размещенный в репозитории. Для каждой работы должны быть доступны исходный код, инструкция по запуску, контрольные примеры или тесты и история изменений.

При затруднении рекомендуется последовательность: воспроизвести проблему; минимизировать пример; прочитать сообщение об ошибке; проверить документацию; локализовать участок кода отладчиком или логированием; сформулировать гипотезу; внести изменение; повторно запустить тесты.

Использование ИИ допускается и целенаправленно включается в отдельные задания. Студент обязан понимать внесенный код, проверять его работоспособность, безопасность и соответствие требованиям, а также фиксировать существенные случаи использования ИИ в журнале проекта.

Сгенерированный ИИ код не принимается как доказательство освоения результата, если студент не может объяснить алгоритм, структуру решения, назначение ключевых конструкций и способ проверки корректности.

При выполнении командных упражнений каждый студент должен иметь идентифицируемый вклад в историю репозитория и уметь объяснить изменения, выполненные им лично.

При выборе языка и среды разработки преподаватель согласует инструментарий с основной дисциплиной «Программирование». Допускается использование современных IDE, Git, средств автоматического форматирования и статического анализа, фреймворков модульного тестирования и доступных ИИ-ассистентов разработчика.

4.5. Примерная тематика курсовых работ (проектов)

Курсовая работа учебным планом не предусмотрена. В рамках дисциплины выполняются два практических проекта (по одному в каждом семестре). Возможные темы: менеджер задач; обработчик журналов событий; конвертер и валидатор данных; консольная информационная система; анализатор текстовых файлов; учебный сервис обработки данных; автоматизация рутинной операции пользователя. Тема конкретизируется преподавателем либо выбирается студентом по согласованию.

V. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) Основная литература

- Документация выбранного языка программирования и его стандартной библиотеки (актуальная версия).
- Chacon S., Straub B. Pro Git. Электронное издание, доступное на официальном сайте Git.
- Документация используемого фреймворка модульного тестирования и среды разработки.

б) Дополнительная литература

- Мартин Р. Чистый код: создание, анализ и рефакторинг.
- Фаулер М. Рефакторинг: улучшение существующего кода.
- Материалы по алгоритмам и структурам данных, рекомендованные преподавателем основной дисциплины «Программирование».

в) Периодическая литература

Профессиональные статьи и инженерные публикации по разработке программного обеспечения, тестированию, Git и применению ИИ в программировании.

г) Базы данных, информационно-справочные и поисковые системы

- официальная документация выбранного языка и библиотек;

- репозитории и документация Git / GitLab / GitHub / GitVerse / SourceCraft — в зависимости от используемой инфраструктуры;
- документация IDE и средств тестирования/статического анализа;
- доступные образовательной организации ИИ-инструменты для генерации и анализа кода.

VI. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-лабораторное оборудование

Компьютерный класс с рабочими местами обучающихся и преподавателя, доступом к локальной сети и Интернету, средствами демонстрации. Требуется возможность установки/запуска среды разработки и работы с удаленным Git-репозиторием.

6.2. Программное обеспечение

- операционная система, используемая в компьютерном классе;
- среда разработки/редактор кода;
- интерпретатор или компилятор выбранного языка;
- Git и клиентские средства работы с репозиториями;
- фреймворк модульного тестирования;
- линтер/форматтер и при необходимости средства статического анализа;
- браузер и инструменты доступа к ИИ-ассистентам, разрешенным образовательной организацией.

6.3. Технические и электронные средства

Система управления обучением, репозиторий учебной группы, средства видеосвязи и демонстрации экрана при использовании электронного обучения и дистанционных образовательных технологий.

VII. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Основу обучения составляет практическая деятельность. Используются problem-based learning, программирование по спецификации, разбор дефектов, парное программирование, code review, мини-проекты, защита решений и техническое собеседование. Преподаватель демонстрирует прием работы на коротком примере, после чего студент применяет его к новой задаче. Для части работ используются открытые условия, допускающие несколько корректных решений, которые необходимо сравнить и обосновать.

ИИ-инструменты включаются только в те задания, где их применение связано с формируемыми результатами ПК-8 ППК-Р7: генерация вариантов реализации и тестов, анализ дефектов, рефакторинг, проверка качества. Обязательными элементами являются верификация, объяснение и документирование результатов.

VIII. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. Оценочные средства текущего контроля

Текущий контроль включает практические задания, проверку репозитория, code review, мини-проект, задания на тестирование и отладку, а также задания на контролируемое использование ИИ. Конкретные задания, критерии и шкалы приведены в ФОС дисциплины.

8.2. Оценочные средства для промежуточной аттестации

Промежуточная аттестация проводится в форме зачета в каждом семестре. Для получения зачета студент должен выполнить обязательный набор практических работ, представить репозиторий с историей изменений и успешно защитить семестровый практический проект. Во 2 семестре защита

дополнительно включает короткое техническое собеседование по коду и демонстрацию корректного использования ИИ-инструментов с объяснением принятых и отклоненных рекомендаций.