Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Выборнова Любовь Алексеевна
Должность: Ректор
Дата подписания: 02.11.2026 10:49:08
Уникальный программный ключ:
c3b3b9c625f6c113afa2a2c42baff9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования «Поволжский государственный университет сервиса»
(ФГБОУ ВО «ПВГУС»)

УТВЕРЖДЕНО

на заседании Высшей школы интеллектуальных систем и кибертехнологий

Протокол от 23.05.2025 № 10

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ
(фонд оценочных средств)
для проведения промежуточной аттестации по дисциплине

«Функциональное программирование»
наименование дисциплины

по образовательной программе высшего образования – программе **бакалавриат**

«Информационные системы и технологии»
наименование образовательной программы

09.03.02 «Информационные системы и технологии»
шифр, наименование направления подготовки / специальности

Составители Коновалова Анжелика Денисовна, старший преподаватель, Высшая школа интеллектуальных систем и кибертехнологий
ФИО, должность, структурное подразделение, ученая степень, ученое звание

Тольятти
2025

1. Паспорт фонда оценочных средств (далее – ФОС)

1.1. Планируемые результаты обучения по дисциплине

Код компетенции	Наименование компетенции	Код и наименование индикаторов достижения компетенции
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	ИОПК-6.1. Применяет методы проектирования программного обеспечения ИОПК-6.2. Использует современные языки программирования для разработки алгоритмов и программ
ПК-2	Способен к выполнению работ по сопровождению и разработке прототипов ИС	ИПК-2.2. Разрабатывает прототип ИС в соответствии с требованиями ИПК-2.3. Осуществляет тестирование прототипа ИС на проверку корректности архитектурных решений

В результате изучения дисциплины обучающийся должен:

Знает: математические основы функционального программирования, понятия чистой функции, неизменяемости, рекурсии, композиции, функций высшего порядка и алгебраических типов данных; принципы оценки результативности программных решений. роль функциональной парадигмы в проектировании программного обеспечения, критерии выбора функциональных языков и библиотек, подходы к декомпозиции предметной области на чистое ядро и слой эффектов.

Умеет: формализовать прикладные задачи в виде композиции функций, применять рекурсивные алгоритмы и преобразования коллекций, анализировать информационные потоки и сравнивать варианты реализации по корректности, сложности и ресурсным показателям. выбирать функциональные средства и технологии для проектируемого решения, разрабатывать концептуальную модель прикладной области с использованием неизменяемых данных и композиции функций, планировать реализацию функционального модуля

Владеет: навыками разработки, отладки и тестирования функциональных программ и функциональных модулей прикладных информационных систем, а также приемами инженерной оценки полученных решений; проектирования структуры функционального программного решения, документирования проектных решений и разработки прототипа прикладного модуля в соответствии с заданными требованиями.

1.2. Содержание дисциплины

№	Тема (раздел дисциплины) (в соответствии с РПД)	Код компетенции
1.	ФУНКЦИОНАЛЬНАЯ ПАРАДИГМА И МАТЕМАТИЧЕСКИЕ ОСНОВЫ	ОПК-6 ПК-2
2.	ВЫРАЖЕНИЯ, ТИПЫ И НЕИЗМЕНЯЕМЫЕ ДАННЫЕ	ОПК-6 ПК-2
3.	ФУНКЦИИ ВЫСШЕГО ПОРЯДКА, КОМПОЗИЦИЯ И КАРРИРОВАНИЕ	ОПК-6 ПК-2
4.	РЕКУРСИЯ И РЕКУРСИВНЫЕ СТРУКТУРЫ ДАННЫХ	ОПК-6 ПК-2
5.	АЛГЕБРАИЧЕСКИЕ ТИПЫ ДАННЫХ И СОПОСТАВЛЕНИЕ С ОБРАЗЦОМ	ОПК-6 ПК-2
6.	ЛЕНИВЫЕ ВЫЧИСЛЕНИЯ И ПОСЛЕДОВАТЕЛЬНОСТИ	ОПК-6 ПК-2
7.	ФУНКЦИОНАЛЬНЫЕ АБСТРАКЦИИ: ФУНКТОРЫ, АППЛИКАТИВЫ И МОНАДИЧЕСКИЕ ВЫЧИСЛЕНИЯ	ОПК-6 ПК-2
8.	ФУНКЦИОНАЛЬНАЯ АРХИТЕКТУРА ПРИКЛАДНЫХ СИСТЕМ	ОПК-6 ПК-2
9.	ПРОЕКТИРОВАНИЕ И ТЕСТИРОВАНИЕ ФУНКЦИОНАЛЬНЫХ ПРОГРАММ	ОПК-6 ПК-2

1.3. Система оценивания по дисциплине

Дисциплина изучается в течении одного семестра.
Форма промежуточной аттестации по дисциплине – экзамен.

Уровень сформированности	100-балльная шкала, %	Шкала освоения, %	5-балльная дифференцированная оценка/балл
допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2
пороговый	61–85,9	61–69,9	«удовлетворительно» / 3
пороговый	61–85,9	70–85,9	«хорошо» / 4
повышенный	86–100	86–100	«отлично» / 5

2. Перечень оценочных материалов для проведения промежуточной аттестации

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в ходе текущего контроля успеваемости (в процессе проведения практических занятий, тестирования, опросов).

В ходе проведения промежуточной аттестации осуществляется контроль и оценка результатов освоения компетенций.

Перечень вопросов и заданий для подготовки к экзамену

ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий

1. Понятие функциональной парадигмы программирования. Ее отличия от императивной и объектно-ориентированной парадигм.
2. Чистые функции и ссылочная прозрачность. Их значение для надежности и тестируемости программ.
3. Функции первого класса. Передача функции как аргумента и возврат функции как результата.
4. Лямбда-исчисление как концептуальная основа функционального программирования.
5. Вывод типов и параметрический полиморфизм в функциональных языках.
6. Неизменяемые данные: преимущества, ограничения и способы обновления структур через создание новых значений.
7. Списки, кортежи и записи в F#: назначение и типовые операции.
8. Функции высшего порядка: назначение и примеры применения.
9. Каррирование и частичное применение функций.
10. Композиция функций и оператор конвейера в F#.
11. Функции map, filter, choose и fold: семантика и области применения.
12. Fold как обобщенная схема свертки структуры данных.
13. Рекурсия: базовый случай, рекурсивный шаг и условия завершения.
14. Хвостовая рекурсия и ее оптимизация.
15. Рекурсивные типы данных. Пример представления дерева.
16. Оценка вычислительной сложности рекурсивных алгоритмов.
17. Алгебраические типы данных и дискриминируемые объединения.
18. Сопоставление с образцом. Исчерпывающий разбор вариантов.
19. Option как безопасное представление отсутствующего значения.
20. Result как способ явного представления успешного и ошибочного результата.
21. Моделирование состояний бизнес-объекта через систему типов.
22. Строгие и ленивые вычисления. Основные различия.
23. Последовательности seq в F# и отложенное выполнение.
24. Бесконечные последовательности: принципы построения и безопасного использования.
25. Функтор как абстракция применения функции к значению в контексте.
26. Аппликативный стиль для комбинирования независимых вычислений.

27. Монадическая композиция. Операция bind и ее назначение.
28. Вычислительные выражения F#: назначение и примеры.
29. Композиция вычислений с Option и Result.
30. Асинхронные вычисления в функциональном стиле.
31. Побочные эффекты и способы их локализации.
32. Архитектурный принцип «functional core, imperative shell».

ПК-2.. Способен к выполнению работ по сопровождению и разработке прототипов ИС

33. Побочные эффекты и способы их локализации.
34. Архитектурный принцип «functional core, imperative shell».
35. Организация ввода-вывода в функциональном приложении.
36. Функциональная декомпозиция бизнес-логики прикладной информационной системы.
37. Критерии выбора функционального языка и технологического стека для прикладной задачи.
38. Планирование работ по разработке функционального модуля.
39. Концептуальная модель прикладной области и ее отражение в системе типов.
40. Модульное тестирование чистых функций.
41. Property-based testing: идея свойств, генерация данных и инварианты.
42. Тестирование функций с эффектами через выделение чистого ядра.
43. Рефакторинг императивного кода в функциональный стиль.
44. Профилирование функциональной программы и оценка ресурсных показателей.
45. Преимущества и ограничения функционального подхода при разработке корпоративных приложений.
46. Интеграция функционального модуля с объектно-ориентированным или императивным кодом.
47. Документирование архитектурных решений функционального программного обеспечения.
48. Типовые практические задания при промежуточной аттестации
49. Преобразовать заданный императивный алгоритм обработки коллекции в функциональный конвейер и объяснить выбор map/filter/fold.
50. Спроектировать типы данных для прикладной области, исключив некорректные состояния с помощью дискриминируемых объединений.
51. Разработать цепочку валидации входных данных с использованием Result и композиции функций.
52. Спроектировать функциональное ядро модуля прикладной ИС и обозначить границы побочных эффектов.
53. Разработать набор модульных и property-based тестов для заданной чистой функции.
54. Выбрать функциональную технологию для заданного сценария и обосновать решение по критериям требований, рисков и трудоемкости.
55. Построить план разработки функционального модуля и представить концептуальную модель предметной области.
56. По результатам профилирования сравнить две функциональные реализации и предложить оптимизацию без нарушения корректности.