

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Крюков Вадим Николаевич

Должность: Проректор по образовательной деятельности и инновационной политике

Дата подписания: 17.06.2026 18:16:14

Уникальный программный ключ:

1b0adb7fd710f6a0705d90c58682bd0c5f2f25b2

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Заполярный государственный университет им. Н. М. Федоровского»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
по дисциплине
Технологии программирования

Направление подготовки 09.03.03 «Прикладная информатика»
Профиль «Прикладная информатика в экономике»
Уровень образования: бакалавриат

Кафедра «Информационные системы и технологии»

Разработчик ФОС:

кандидат экономических наук, доцент, И.С. Беляев _____

И.С. Беляев

Оценочные материалы по дисциплине рассмотрены и одобрены на заседании
кафедры, протокол от 10.04.2026г. № 5.

Заведующий кафедрой _____ к.э.н., Беляев И.С.

Фонд оценочных средств по дисциплине Технологии программирования для текущей/ промежуточной аттестации разработан в соответствии с Федеральным государственным образовательным стандартом высшего образования по специальности / направлению подготовки 09.03.03 Прикладная информатика на основе Рабочей программы дисциплины Технологии программирования, утвержденной решением ученого совета от г., Положения о формировании Фонда оценочных средств по дисциплине (ФОС), Положения о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся ЗГУ, Положения о государственной итоговой аттестации (ГИА) выпускников по образовательным программам высшего образования в ЗГУ им. Н.М. Федоровского.

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами образовательной программы

Таблица 1. Компетенции и индикаторы их достижения

Код и наименование компетенции	Индикаторы достижения
ОПК-1 Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;	ОПК-1.1 Понимает основы математики, физики и информатики
	ОПК-1.2 Формулирует решение стандартных профессиональных задач с применением естественнонаучных и общеинженерных знаний
	ОПК-1.3 Применяет методы теоретического и экспериментального исследования для решения практических задач профессиональной деятельности
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения;	ОПК-7.1 Осваивает современные языки программирования и программные среды для разработки программ, пригодных для практического применения
	ОПК-7.2 Применяет языки программирования, современные программные среды для разработки и сопровождения программ, пригодных для практического применения

	ОПК-7.3 Использует навыки алгоритмизации, программирования, отладки и тестирования информационных систем
--	--

ПК-1 Способность разрабатывать и адаптировать прикладное программное обеспечение	ПК-1.1 Использует основы организационного и технологического обеспечения процесса разработки прикладного программного обеспечения, принципы оптимизации и рефакторинга программного кода
	ПК-1.2 Формализует и проводит алгоритмизацию прикладных задач, разрабатывает структуру программы и решения по интеграции ее модулей
	ПК-1.3 Выполняет написание, отладку и оптимизацию программного кода, осуществляет интеграцию программных модулей и
ПК-6 Способность проводить тестирование компонентов программного обеспечения ИС	ПК-6.1 Использует методы и приемы тестирования экономической информационной системы и ее компонентов
	ПК-6.2 Разрабатывает и выбирает программы тестирования компонентов экономической информационной системы
	ПК-6.3 Выполняет тестирование компонентов экономической информационной системы с помощью разработанных и стандартных программных средств

Таблица 2. Паспорт фонда оценочных средств

№п/п	Контролируемые разделы(темы) дисциплины	Кодрезультатаобучения по дисциплине/ модулю	Оценочные средства текущей		Оценочные средства промежуточной	
			Наименование	Форма	Наименование	Форма
6 семестр						
7 семестр						

2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие процесс формирования компетенций в ходе освоения образовательной программы.

2.1. Задания для текущего контроля успеваемости

1. Понятие информационной системы. Виды информационных систем.
2. Классификация информационных систем по степени оснащенности аналитическими инструментами.
3. Функции системного интегратора. Задачи, выполняемые системными аналитиками, системными программистами и прикладными программистами.
4. Понятие алгоритма. Классификация алгоритмов.
5. Виды технологий программирования: их особенности и различия.
6. Понятие интегрированной системы программирования. Составные части интегрированной системы программирования, их функции.
7. Определение классов и объектов. Инкапсуляция.
8. Наследование методов: перекрытие и виртуализация.
9. Области видимости объектов.
10. Структура программ Delphi. Файл проекта.
11. Визуальные и не визуальные компоненты.
12. Компоненты страницы Standard.
13. Компоненты страницы Additional.

14. Компоненты страницы System.

15. Просмотр растровых файлов в Delphi.
16. Формирование окна заставки.
17. Обработка исключений: класс Exception.
18. Динамическое подключение формы.
19. Модальные и немодальные формы.
20. Графические элементы: понятие канвы, ее свойства и методы.
21. Восстановление недействительной части формы.
22. Рекурсия. Примеры вызова рекурсивных подпрограмм.
23. Построение графика затухающего синусоидального сигнала на канве формы.
24. Предварительная обработка нажатия клавиши.
25. Замена клавиш, установленных по умолчанию.
26. Использование компонента Timer.
27. Типы модулей в Delphi.
28. Особенности разработки динамических библиотек.
29. Процессы и субпроцессы. Подключение внешнего приложения.
30. Особенности сохранения программных модулей в Delphi.

1. История развития языков программирования высокого уровня.
2. Архитектура ЯП (3 поколения).
3. Архитектура языков ООП.
4. Сложность, присущая ПО (четыре основные причины).
5. Трудности по созданию сложных систем.
6. Структура сложных систем (5 признаков). Примеры сложных систем (выделить в них признаки).
7. Типовая и структурная иерархии в OOD.
8. Методы ограничения и контроля сложности ПО при использовании различных методов программирования (SD, DD, OOD).
9. Основные понятия: метод, методология, технология. Классификация методов программирования (три группы), общая характеристика их.
10. Эволюция развития программного продукта. Основные определения, понятия, отличительные черты.
11. Понятие "модуль" в программировании. Различные виды модулей при использовании основных методов программирования.
12. Жизненный цикл ПО (ЖЦ). Фазы ЖЦ, их характеристика.
13. Этапы разработки ПО. Различные модели.
14. Краткая характеристика этапов (содержание) разработки ПО при использовании структурных методов.
15. ЖЦ разработки ПО при ОО подходе. Изменения в процессе эволюции программных систем, стоимость каждого вида изменения (в смысле затрат).
16. Документирование программного продукта. Различные виды документов, их содержание.
17. Виды документов при OOD.
18. Этапы разработки ПО согласно ГОСТ 19.102-77 и соответствие их традиционным этапам разработки.
19. Временные затраты на реализацию этапов разработки ПО. Особенности распределения ресурсов при OOD.
20. Управление проектом и контроль за качеством программного продукта.
21. Становление и методология структурных методов разработки. Структурное программирование.
22. Способы (системы обозначений) описания логики модулей структурных

ме-тодов разработки.

2.2 Темы письменных работ (эссе, рефераты, курсовые работы и др.)

1. Разработка приложений

1. Разработка кроссплатформенного мобильного приложения с использованием Flutter.

2. Создание веб-приложения для управления задачами на базе React.js.

3. Разработка системы управления складскими запасами с использованием Java и Spring Boot.

4. Создание десктопного приложения для анализа данных с использованием Python и PyQt.

5. Разработка системы рекомендаций для онлайн-магазина на базе машинного обучения.

6. Создание RESTful API для управления базой данных пользователей.

7. Разработка чат-бота для Telegram с использованием Python.

8. Создание приложения для анализа текстовых данных с использованием NLP (Natural Language Processing).

9. Разработка системы управления проектами с использованием ASP.NET Core.

10. Создание игрового приложения с использованием Unity.

2. Алгоритмы и структуры данных

11. Реализация алгоритма сортировки слиянием и его сравнение с быстрой сортировкой.

12. Разработка программы для решения задачи о рюкзаке с использованием динамического программирования.

13. Создание алгоритма для поиска кратчайшего пути в графе (алгоритм Дейкстры).

14. Реализация алгоритма поиска подстроки в строке (алгоритм Кнута-Морриса-Пратта).

15. Разработка программы для работы с хэш-таблицами и их оптимизации.

16. Создание системы для анализа производительности различных алгоритмов сортировки.

17. Разработка программы для генерации и обработки больших массивов данных.

18. Реализация алгоритма построения минимального остовного дерева (алгоритм Прима).

19. Создание программы для работы с бинарными деревьями поиска.

20. Разработка алгоритма для решения задачи коммивояжёра методом ветвей и границ.

3. Базы данных и работа с данными

21. Разработка системы управления базой данных для интернет-магазина.

22. Создание программы для миграции данных между различными СУБД.

23. Реализация системы аналитики данных с использованием SQL и Python.

24. Разработка системы для автоматизации создания отчетов из базы данных.

25. Создание приложения для работы с NoSQL базами данных (например, MongoDB).

26. Реализация системы для анализа больших данных с использованием Hadoop.

27. Разработка программы для импорта и экспорта данных в формате

JSON/XML.

28. Создание системы для анализа логов сервера с использованием базы данных.

29. Реализация системы для работы с облачными базами данных (например, AWS DynamoDB).

30. Разработка приложения для визуализации данных из базы данных.

4. Искусственный интеллект и машинное обучение

31. Разработка системы прогнозирования цен на акции с использованием машинного обучения.

32. Создание программы для классификации изображений с использованием TensorFlow.

33. Реализация системы для анализа тональности текста с использованием NLP.

34. Разработка системы рекомендаций фильмов на основе пользовательских предпочтений.

35. Создание программы для распознавания объектов на изображениях с использованием OpenCV.

36. Реализация модели машинного обучения для предсказания погоды.

37. Разработка системы для анализа данных с датчиков IoT с использованием AI.

38. Создание программы для генерации текста с использованием GPT-подобных моделей.

39. Реализация системы для анализа поведения пользователей на сайте.

40. Разработка программы для автоматической генерации музыки с использованием нейросетей.

5. Веб-разработка

41. Разработка многостраничного веб-сайта с использованием HTML, CSS и JavaScript.

42. Создание адаптивного веб-приложения для управления персоналом компании.

43. Реализация системы авторизации и аутентификации пользователей на базе JWT.

44. Разработка веб-приложения для онлайн-обучения с использованием Django.

45. Создание системы для управления контентом (CMS) с использованием PHP и MySQL.

46. Реализация системы онлайн-чатов с использованием WebSocket.

47. Разработка веб-приложения для управления финансами пользователя.

48. Создание системы для анализа поведения пользователей на веб-сайте.

49. Реализация системы для работы с API сторонних сервисов (например, Google Maps).

50. Разработка веб-приложения для планирования задач команды.

6. Инструменты разработки и автоматизация

51. Разработка системы автоматизации тестирования веб-приложений.

52. Создание программы для автоматизации сборки проектов с использованием Jenkins.

53. Реализация системы для анализа кода на наличие ошибок с использованием статических анализаторов.

54. Разработка инструмента для автоматизации деплоя приложений в облако.

55. Создание программы для генерации документации к исходному коду.

56. Реализация системы для работы с контейнерами Docker.

57. Разработка инструмента для мониторинга производительности приложений.

58. Создание программы для автоматизации работы с Git.

59. Реализация системы для анализа логов приложений в реальном времени.

60. Разработка инструмента для оптимизации производительности базы данных.

1. Текущий контроль знаний

Тестовые вопросы

1. Что такое алгоритм? Приведите пример.
2. Какие основные парадигмы программирования существуют? (например, объектно-ориентированное, функциональное)
3. В чем заключается разница между компилятором и интерпретатором?
4. Что такое структуры данных? Приведите примеры (например, массивы, списки, деревья).
5. Как работает система контроля версий Git? Приведите основные команды.
6. Что такое рекурсия? Приведите пример рекурсивного алгоритма.
7. Как организуется работа с базами данных в программировании?
8. Что такое API? Где оно применяется?
9. Какие основные принципы ООП вы знаете? (инкапсуляция, наследование, полиморфизм)

10. Что такое регулярные выражения? Для чего они используются?

Практические задания

11. Напишите программу для сортировки массива методом пузырька.
12. Реализуйте функцию для вычисления факториала числа с использованием рекурсии.
13. Создайте простую базу данных пользователей с использованием SQL.
14. Разработайте RESTful API для управления данными о студентах.
15. Напишите скрипт для работы с JSON-файлами.

Открытые вопросы

16. Как искусственный интеллект меняет подходы к программированию?
17. Какие ошибки чаще всего допускаются при написании кода?
18. Как автоматизация помогает в снижении затрат на разработку?
19. Какие факторы влияют на выбор языка программирования для проекта?
20. Как удаленная работа влияет на процессы разработки?

2. Промежуточная аттестация

Тестовые вопросы

21. Что такое микросервисная архитектура? Где она применяется?
22. Какие методы защиты данных используются в программировании?
23. В чем заключается концепция "чистого кода"?
24. Как технологии программирования поддерживают глобализацию?
25. Что такое KPI? Для чего они используются в разработке?

Практические задания

26. Разработайте систему KPI для анализа эффективности программы.
27. Создайте план управления рисками для проекта через программирование.
28. Реализуйте систему уведомлений для пользователей приложения.
29. Настройте интеграцию облачных сервисов с программной системой.
30. Разработайте план миграции данных из одной системы в другую.

Ситуационные задачи

31. Компания хочет автоматизировать процесс обработки данных через программирование. Как вы будете строить модель системы?

32. У вас есть набор данных о поведении пользователей. Как вы будете их анализировать через программирование?

33. Вам нужно разработать систему для прогнозирования рыночных трендов. Какие технологии вы выберете?

34. Вы работаете над проектом управления рисками. Как технологии программирования могут быть применены в этой задаче?

35. Компания хочет автоматизировать процесс анализа данных. Как вы будете использовать программирование для решения этой задачи?

3. Итоговый контроль

Защита курсового проекта

36. Разработка системы управления проектами с использованием современных технологий.

37. Проектирование системы анализа данных для принятия решений.

38. Создание системы управления складскими запасами через программирование.

39. Разработка системы рекомендаций для клиентов интернет-магазина.

40. Проектирование системы управления качеством продукции через программирование.

Экзамен

41. Теоретический блок: тестовые вопросы по основным темам курса.

42. Практический блок: выполнение задания на разработку программы.

43. Анализ предоставленного кода и исправление ошибок.

Оценка портфолио

44. Сборник выполненных лабораторных работ.

45. Документация по разработанным проектам.

46. Отчеты по практическим заданиям.

4. Дополнительные виды оценочных средств

47. Устное собеседование по ключевым темам курса.

48. Анализ кода и исправление ошибок в предоставленном проекте.

49. Выполнение заданий на оптимизацию производительности программы.

50. Разработка прототипа системы управления процессом с использованием современных технологий.

1. Текущий контроль знаний:

- Тесты на знание основ программирования (например, парадигмы, структуры данных, алгоритмы).
- Практические задания по написанию и отладке кода (например, реализация алгоритмов или работа с базами данных).
- Написание скриптов для обработки данных или автоматизации процессов.
- Лабораторные работы по созданию и тестированию программных модулей.

2. Промежуточная аттестация:

- Выполнение контрольной работы (например, разработка программы для решения конкретной задачи).
- Рефераты или эссе на заданные темы (например, "Роль объектно-ориентированного программирования в современных технологиях").
- Мини-проект (например, разработка прототипа системы управления

задачами или базы данных).

3. Итоговый контроль:

- Защита курсового проекта (разработка полноценной программной системы или её компонентов).

- Экзамен в виде теста или выполнения практического задания.

- Оценка портфолио выполненных лабораторных и практических работ.

4. Дополнительные виды оценочных средств:

- Устное собеседование по ключевым темам курса.

- Анализ кода и исправление ошибок в предоставленном проекте.

- Выполнение заданий на оптимизацию производительности программы.