Энгельсский технологический институт (филиал) федерального государственного бюджетного образовательного учреждения высшего образования

«Саратовский государственный технический университет имени Гагарина Ю.А.»

Кафедра «Естественные и математические науки»

Оценочные материалы по практике

Б.2.1.2 «Учебная технологическая (проектно-технологическая) практика»

направления подготовки

09.03.04 «Программная инженерия»

профиль

«Управление разработкой программных проектов»

Энгельс 2026

1. Перечень компетенций и уровни их сформированности по дисциплинам (модулям), практикам в процессе освоения ОПОП ВО

В процессе освоения образовательной программы обучающийся проходит учебную технологическую (проектно-технологическую) практику, целью которой является отработка компетенции ОПК-2 и ОПК-6, которые были ранее сформированы в процессе изучения соответствующих дисциплин учебного плана:

Код и наименование индикатора достижения компетенции	Дисциплина, в результате формируемся компетенция
ИД-1$_{ОПК-6}$	Б.1.1.9 Программирование
ИД-2$_{ОПК-6}$ ИД-2$_{ОПК-2}$	Б.1.1.11 Базы данных
ИД-1$_{ОПК-2}$	Б.1.1.10 Информатика

Критерии определения сформированности компетенций на различных уровнях их формирования

Индекс компетенции	Содержание компетенции
ОПК-2	Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности

Код и наименование индикатора достижения компетенции	Виды занятий для формирования компетенции	Оценочные средства для оценки уровня сформированности компетенции
ИД-1$_{ОПК-2}$ Знает и понимает основные принципы современных информационных технологий и может их использовать при решении практических задач	Практическая работа, самостоятельная работа	вопросы для проведения зачёта, тестовые задания
ИД-2$_{ОПК-2}$ Знает и понимает основные понятия и методы проектирования баз данных, основы языка запросов SQL к реляционным БД для решения задач профессиональной деятельности	Практическая работа, самостоятельная работа	вопросы для проведения зачёта, тестовые задания

Уровни освоения компетенции ИД-1$_{ОПК-2}$

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	Знает: современные среды разработки ПО. Умеет: применять современные среды разработки ПО для решения практических задач. Владеет: навыками разработки ПО в различных средах разработки.
Повышенный (хорошо)	Знает: современные среды разработки ПО. Умеет: применять одну или несколько сред разработки ПО для решения практических задач. Владеет: навыками разработки ПО в одной или нескольких средах разработки.
Пороговый (базовый) (удовлетворительно)	Знает: одну или несколько сред разработки ПО. Умеет: применять одну среду разработки ПО для решения практических задач. Владеет: навыками разработки ПО в одной среде разработки.

Уровни освоения компетенции ИД-2$_{ОПК-2}$

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	Знает: • Глубокое, всестороннее и безошибочное знание всех основных понятий и методов проектирования реляционных БД. • Свободное владение терминологией (инфологическая, логическая, физическая модели, нормализация, связи, ключи). • Четкое понимание теоретических основ и синтаксиса языка SQL. Умеет: • Студент самостоятельно и безошибочно проектирует реляционную базу данных: строит корректную инфологическую (ER-диаграмму), логическую (с учетом нормальных форм) и физическую модели. • Безошибочно и оптимально составляет SQL-запросы любой сложности (включая соединения JOIN, подзапросы, агрегатные функции, группировки). Владеет: Демонстрирует устойчивые навыки комплексного решения профессиональных задач: обоснованно выбирает архитектуру БД под конкретную предметную область, аргументирует проектные решения, пишет эффективные и чистые SQL-запросы.
Повышенный (хорошо)	Знает: • Твердое знание основных понятий и методов проектирования и основ SQL, но с незначительными неточностями (например, путаница в деталях нормализации или редких типах связей), которые студент исправляет сам после наводящего вопроса. Умеет: Уверенное проектирование БД и составление SQL-запросов, но допущены 1–2 мелкие неточности (например, избыточность в связях или неоптимальный, но рабочий SQL-запрос), не приводящие к нарушению логики работы БД.

	Владеет: Владеет необходимыми навыками решения профессиональных задач, но испытывает затруднения при переносе стандартных алгоритмов на нестандартную или усложненную предметную область; решения не всегда оптимальны.
Пороговый (базовый) (удовлетворительно)	Знает: • Базовые знания основных понятий проектирования БД и SQL присутствуют, но фрагментарные. • Допускаются грубые ошибки в определении ключевых терминов, путаница между уровнями моделей данных, ответить на дополнительные вопросы преподавателя студент затрудняется. Умеет: • Задание выполнено с существенными методическими или синтаксическими ошибками. • Проектирование выполнено поверхностно (нарушены правила нормализации, неверно указаны типы данных или связи). • SQL-запросы составляются только при помощи преподавателя или содержат грубые синтаксические/логические ошибки. Владеет: • Навыки продемонстрированы на минимальном уровне; решение профессиональных задач вызывает значительные затруднения, студент способен справиться только с простейшими шаблонными задачами репродуктивного характера.

Индекс компетенции	Содержание компетенции
ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов.

Код и наименование индикатора достижения компетенции	Виды занятий для формирования компетенции	Оценочные средства для оценки уровня сформированности компетенции
ИД-1$_{ОПК-6}$ Применяет основы информатики и программирования для разработки алгоритмов и программ, пригодных для практического использования	Практическая работа, самостоятельная работа	вопросы для проведения зачёта, тестовые задания

ИД-2$_{ОПК-6}$ Разрабатывает базы данных средствами СУБД.	Практическая работа, самостоятельная работа	вопросы для проведения зачёта, тестовые задания

Уровни освоения компетенции ИД-1$_{ОПК-6}$

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	Умеет: писать и читать исходный текст программ на языке C# повышенной сложности с использованием различных библиотек функций, комментировать текст программы, компилировать, запускать программу и отлаживать программу. Владеет: навыками применения интегрированной среды разработки для создания приложений на языке C# повышенной сложности с учетом особенностей среды разработки
Повышенный (хорошо)	Умеет: писать и читать исходный текст программ на языке C# средней сложности с использованием различных библиотек функций, комментировать текст программы, компилировать, запускать программу и отлаживать программу Владеет: навыками применения интегрированной среды разработки для создания приложений на языке C# средней сложности
Пороговый (базовый) (удовлетворительно)	Умеет: писать и читать исходный текст простейших программ на языке C#, компилировать и запускать программу Владеет: навыками применения интегрированной среды разработки для создания простейших приложений на языке C#

Уровни освоения компетенции ИД-2$_{ОПК-6}$

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	Умеет: разрабатывать хранимые процедуры и функции для решения поставленных задач. Владеет: навыками проектирования и физической реализации реляционных баз данных средствами СУБД, навыками поддержания целостности и корректности данных с использованием хранимых процедур и функций;.
Повышенный (хорошо)	Умеет: разрабатывать триггеры, хранимые процедуры и функции для решения поставленных задач. Владеет: навыками проектирования и физической реализации реляционных баз данных средствами СУБД, навыками поддержания целостности и корректности данных с использованием триггеров, хранимых процедур и функций.
Пороговый (базовый) (удовлетворительно)	Умеет: разрабатывать триггеры, хранимые процедуры и функции для решения поставленных задач, в том числе и для баз данных большого объема.

	Владеет: навыками проектирования и физической реализации реляционных баз данных средствами СУБД, навыками поддержания целостности и корректности данных с использованием триггеров, хранимых процедур и функций; навыками представления данных в удобном формате с использованием хранимых процедур и функций.

2. Методические, оценочные материалы и средства, определяющие процедуры оценивания сформированности компетенций (элементов компетенций) в процессе освоения ОПОП ВО

2.1 Оценочные средства для текущего контроля

Задание на практику

Тема 1: «РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ»

Разработать игровое приложение (тема согласовывается с преподавателем). Игровое приложение должно иметь: оформленную концепцию (описание игрового мира, героев, цель игры), несколько уровней сложности, подсчет игровых очков, рейтинг лучших игроков, регистрация игроков, систему обучения, подсказок и помощи игрокам.

Приложение должно иметь базу данных, графический интерфейс (веб – приложение или настольное приложение). В качестве языка программирования необходимо выбрать один из изученных языков (C, C#, JavaScript, Java). Среда разработки будущего приложения студентом определяется самостоятельно.

Тема 2: «РАЗРАБОТКА ВЕБ-САЙТА»

Разработать динамический веб-сайт (тема согласовывается с преподавателем).

Тема 3: «РАЗРАБОТКА ИНЖЕНЕРНОГО ПРИЛОЖЕНИЯ»

Разработать инженерное приложение по чледующим подтемам: (тема согласовывается с преподавателем).

1. Провести расчет индуктивности многослойной катушки, имеющей размеры dmin (минимальный диаметр), dmax (максимальный диаметр) ,b (ширина) и w (число витков).
2. Рассчитать параметры сварного шва двух полос из стали марки СТ-3. Составить схему алгоритма, выбирающую тип сварного шва (угловой или стыковой) в зависимости от исходных данных: величины статической растягивающей нагрузки P , толщины a и ширины b стальных полос. При сварке полос из стали СТ3 использованы электроды Э34.
3. Провести расчет зависимости величины предельной растягивающей нагрузки P для сварного стыкового шва от его длины b' для двух стальных полос, толщиной a
4. Составить схему алгоритма расчёта результирующего изгибающего момента на валу при заданных нагрузках Ft и Fr и определённой длине l.
5. Расчет зависимости изгибающего момента на вал от нагрузок. Используя формулы, произвести расчет зависимости изгибающего момента на вал от нагрузок, которые изменяются в определенных пределах:

6. Составить схему алгоритма расчёта конденсатора C, сглаживающего пульсации выпрямленного тока в условиях изменяющейся нагрузки R

7. Составить схему алгоритма расчета сопротивления R проводов из 10 различных материалов, используя формулу:

8. Составить схему алгоритма определения длины провода диаметра d, который используется для изготовления реостата, имеющего сопротивление R.

9. Произвести расчет делителя напряжения цепи постоянного тока.

В качестве языка программирования необходимо выбрать один из изученных языков (C, C#, JavaScript, Java). Среда разработки будущего приложения студентом определяется самостоятельно.

2.2 Оценочные средства для промежуточного контроля

Вопросы к зачету

Зачет по практике проходит в форме собеседования, предметом которого является разработанное приложение, этапы и особенности его разработки. Компетенции считаются сформированными, если студент разработал приложение, подготовил отчет по проделанной работе и успешно прошел собеседование.

Список вопросов зависит от языка и среды разработки. Ниже приведен примерный список вопросов, которые могут быть заданы на собеседовании.

1. Операторы. Приоритеты операторов. Бинарные и унарные операторы.
2. Выражения. Выражения и преобразование арифметических типов. Преобразование типов.
3. Основные компоненты текста программы на языке C. Функции, описания, директивы препроцессора, комментарии. Подключениевнешних библиотек. Компилирование и выполнение программы.
4. Заголовочный файл <stdio.h>. Printf(). Модификаторы форматных команд в printf(). Scanf(). Буферизированный ввод-вывод. Getchar(). Putchar().
5. Понятие алгоритма. Свойства алгоритмов. Способы записи алгоритмов.
6. Алгоритмические конструкции – следование, ветвление, цикл. Теорема структуры и структурное программирование.
7. Представление основных управляющих структур программирования в языке C. Структура следования. Структура ветвления.
8. Операторы выбора if и switch. Конструкция ‘if’. Конструкция ‘if…else’. Конструкция ‘else - if’. Вложенные операторы if. Оператор switch.
9. Цикл 'for'. Цикл 'while'. Цикл 'do...while'. Вложенные циклы.
10. Операторы безусловного перехода.
11. Определение массива в языке C. Элементы и индексы массива. Работа с массивами в C.
12. Многомерные массивы. Инициализация массива.
13. Строки/символьные массивы. Строковые переменные и константы. Операции вводавывод строк. Функции стандартной библиотеки языка C для работы со строками (string.h)
14. Указатели и адреса в языке C.Операция получения адреса &. Переменные типа указатель. Доступ к переменной по указателю.
15. Арифметические операции с указателями. Сравнение указателей.
16. Указатели и одномерные массивы.
17. Указатели и многомерные массивы.

18. Структуры и их использование. Определение структур. Объявление структурных переменных.
19. Массивы структур. Указатели на структуры.
20. Потоки и файлы. Функции стандартной библиотеки языка С для работы с файлами (stdio.h). Файловый указатель. Аргументы командной строки. Открытие и закрытие файла. Чтение из файл. Запись в файл.
21. Модульные программы. Принцип модульности программы. Понятие подпрограммы. Критерии создания подпрограмм. Свойства подпрограмм. Процедуры и функции.
22. Особенности написания и использования функций в языке С. Прототипы и определения.
23. Параметры функций. Передача параметров по значению. Передача параметров по ссылке. Возвращаемое значение. Структуры как аргументы функций.
24. Что представляет собой язык HTML. Спецификация HTML.
25. Описание раздела заголовка HTML-документа.
26. Физическое и логическое форматирование текста HTML-документа.
27. Форматирование текстовых блоков.
28. Цветовые модели и форматы хранения графики для веб.
29. Способы добавления графики на веб-страницу.
30. Организация гиперссылок: внешней, локальной, внутренней.
31. Назначение, виды и способ организации карты-изображения в HTML-документе.
32. Преимущества и недостатки табличного способа представления данных в HTMLдокументе.
33. Организации таблицы в в HTML-документе.
34. Преимущества и недостатки использования фреймовой структуры для представления данных на веб-странице.
35. Организация веб-страницы на основе фреймов.
36. Стилевое оформление веб-страниц. Задание свойств стиля.
37. Создание списков: нумерованных, маркированных, списков определений.
38. Создание таблиц. Основные теги и атрибуты.
39. Формы на web-странице, атрибуты формы.
40. Добавление элементов в форму: текстовое поле, поле со списком, радиокнопки, флажки, кнопки, текстовые области, поля с проверкой, цвет, дата и время и т.д., атрибуты элементов формы.
41. Типы данных в Java Script. Свободная типизация данных.
42. Имена переменных. Способы создания переменных. Область действия переменных.

Локальные и глобальные переменные.

43. Ввод и вывод данных. Методы преобразования типов в Java Script.
44. Операторы условного перехода. Оператор выбора, операторы цикла.
45. Функции, встроенные функции, пользовательские функции. Особенности использования переменных в функциях.
46. Где, что и как делают сценарии. Расположение сценариев. Понятие события, обработка событий в Java Script, свойства событий.
47. Понятие коллекций объектов документа. Прохождение событий. Указание обработчика события в сценарии.
48. Объекты в Java Script.
49. Объект window. Свойства window. Методы window. События window.

50. Объект документ. Свойства document. Коллекции document. Методы document. События document.
51. Работа с окнами в Java Script. Создание новых окон. Закрытие окон.
52. Этапы разработки БД. Понятия объекта, предмета, предметной области, информационной модели. Концептуальное проектирование. Логическое проектирование. Физическое проектирование.
53. Проектирование БД методом нормальных форм. Основные нормальные формы, их особенности.
54. Инфологическая модель данных «Сущность-Связь». История развития семантических моделей. Понятие ER-диаграммы.
55. Правила построения ER-диаграмм. Построение связей средствами ER-диаграмм. Понятие сущности и атрибута в ER-модели.
56. Понятие нормальных форм для ER-схем. Сложные элементы ER-модели, их краткая характеристика. Порядок разработки ER-модели.
57. Преобразование ER-диаграммы в реляционную схему (SQL-ориентированную).
58. Основные понятия реляционной теории баз данных. Понятие отношения, атрибута, поля, кортежа, записи. Основные свойства отношений.
59. Связи между отношениями: тип связи; родительская и дочерняя (зависимая) сущность; мощность связи; допустимость пустых (null) значений.
60. Правила обеспечения целостности в реляционной модели: целостность сущностей, целостность ссылок.
61. Стратегии поддержания ссылочной целостности.
62. Понятие триггера. Достоинства и недостатки реляционной модели
63. Основные операции реляционной алгебры. Операция объединения, пересечения, разности, декартово произведения (умножения), ограничения (выборки), проекции, соединения, деления. Понятие совместимости по взятию расширенного декартова произведения. Понятие совместимость отношений по объединению.
64. Реляционное исчисление. Исчисление кортежей. Правильно построенные формулы. Простые условия. Кванторы, свободные и связанные переменные.
65. Реляционное исчисление. Целевые списки и выражения реляционного исчисления. Исчисление доменов. Условие членства. Выражения исчисления доменов. 66. Реализация основных операций в SQL: выборка, вставка, удаление, обновление данных. Операторы определения объектов базы данных (DDL). Их перечень, основные характеристики.
66. Основные принципы объектно-ориентированного программирования. Инкапсуляция, наследование, полиморфизм.
67. Преимущества объектно-ориентированного программирования.
68. Понятие класса и объекта класса.
69. Структура программы на языке C#. Понятия пространства имен. Директива using. Синтаксис описания простейшего класса и метода Main.
70. Классификация типов данных в языке C#: встроенные и определяемые программистом, простые и структурные, типы значения и ссылочные типы.
71. Встроенные типы языка C# .
72. Преобразование типов.

73. Основные методы класса Console и их параметры: Write, WriteLine, Read, ReadLine. Особенности чтения данных с клавиатуры в C#.
74. Общий механизм обработки исключений. Преимущества исключений.
75. Синтаксис исключений. Перехват исключений. Операторы checked и unchecked.
76. Одномерные массивы в языке C#.
77. Двумерные массивы в языке C#.
78. Ступенчатые массивы в языке C#.
79. Класс System.Char.
80. Строки типа String.
81. Строки типа StringBuilder.
82. Синтаксис описания класса в C#. Типы элементов класса. Спецификаторы видимости элементов класса.
83. Синтаксис описания полей и констант класса в языке C#.
84. Синтаксис описания методов класса в языке C#. Перегрузка методов.
85. Ключевое слово this.
86. Конструкторы и деструкторы класса в языке C#.
87. Свойства класса в языке C#.
88. Метод Main в языке C#.
89. Индексаторы в языке C#.
90. Операции класса в языке C#.
91. Описание производного класса в языке C#. Правила наследования элементов класса.
92. Виртуальные методы.
93. Абстрактные классы. Бесплодные классы.
94. Класс object.
95. Синтаксис интерфейса в языке C#. Реализация интерфейса.
96. Работа с объектами через интерфейсы. Операции is, as. Интерфейсы и наследование.
97. Потоки байтов. Потоки символов. Двоичные потоки.
98. Коллекции в C#. Классы-прототипы.
99. Понятие и структура платформы MS.NET.
100. Структура Microsoft.NET Framework.

Таблица 1 – Критерии выставления оценок при проведении текущего контроля и промежуточной аттестации

Шкала оценки	Оценка	Критерий выставления оценки
Четырехбалльная шкала	Отлично	Обучающийся глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, тесно увязывает теорию с практикой. Обучающийся не затрудняется с ответом при видоизменении задания, свободно справляется с задачами, заданиями и другими видами применения знаний, показывает знания законодательного и нормативно-технического материалов, правильно обосновывает принятые решения, владеет разносторонними навыками и приемами выполнения практических работ, обнаруживает умение самостоятельно обобщать и излагать материал, не допуская ошибок
	Хорошо	Обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми навыками при выполнении практических заданий
	Удовлетворительно	Обучающийся усвоил только основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий
	Неудовлетворительно	Обучающийся не выполнил программу практики

2.3. Итоговая диагностическая работа по дисциплине

ЗАДАНИЯ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), ПРАКТИКЕ

2.3. Итоговая диагностическая работа по дисциплине

ЗАДАНИЯ ДЛЯ ДИАГНОСТИЧЕСКОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ), ПРАКТИКЕ

Компетенции:

1.	b	Переменная - это a. значение некоторой области памяти компьютера b. именованная область памяти компьютера, выделяемая для хранения конкретных данных c. адрес поименованной области памяти компьютера, в котором хранится данные определенного типа d. идентификатор, для которого указан тип данных и инициализирующее выражение	ОПК-2
2.	a	По умолчанию локальные переменные имеют класс памяти a. auto b. extern c. static d. register	ОПК-2

3.	a	Какие символы могут использоваться для комментариев? a. /* comment */ b. { comment } c. (* comment *)	ОПК-2
4.	a	Какой из операторов может быть и унарным и бинарным? a. – b. - - c. ++ d. * e. %	ОПК-2
5.	e	Дана последовательность операторов. int a=10, c=20, d=0; printf (“%d %d %d”, ++a, c||d, !d); После ее выполнения на экран будет выведено a. 10 1 1 b. 10 0 1 c. 10 20 0 d. src	ОПК-2
6	Функции позволяют объединять операторы в группы и затем выполнять их, когда это потребуется Функции могут быть переданы аргументы	Какие утверждения о функциях JavaScript являются истинными? 1. Функции позволяют объединять операторы в группы и затем выполнять их, когда это потребуется 2. Функции могут быть переданы аргументы 3. Последним оператором тела функции всегда является оператор return 4. Язык JavaScript не позволяет разработчикам описывать собственные пользовательские функции	ОПК-6.

7	1. SELECT SUM(Price) FROM Sales	7. Какой из следующих запросов выводит сумму по колонке 'Price' из таблицы 'Sales'? 1. SELECT SUM(Price) FROM Sales 2. SELECT TOTAL(Price) FROM Sales 3. SELECT SUM(Price) WHERE Sales 4. SELECT ADD(Price) FROM Sales	ОПК-6
8	1. Содержит два параметра со значениями по умолчанию	8. Данная хранимая процедура: CREATE PROC my_proc5 @t VARCHAR(20)=’Конфеты', @p FLOAT=0.1 AS UPDATE Товар SET Цена=Цена*(1-@p) WHERE Тип=@t 1. Содержит два параметра со значениями по умолчанию 2. Содержит 2 инициализированные переменные 3. Содержит 2 параметра, которые всегда равны значениям Конфеты и 0.01	ОПК-6
9	4. Источники данных для выборки	Секция FROM определяет: 1. Столбцы, по которым будет осуществляться выборка 2. Правила и ограничения для данных 3. Максимальное количество строк в итоговом наборе 4. Источники данных для выборки	ОПК-6

10	3. Список товаров, цена которых больше или равна 100 или меньше или равна 150	Команда SELECT Название, Цена FROM Товар WHERE Цена>=100 And Цена<=150 выполняет следующие действия: 1. Список товаров, цена которых больше или равна 100 и меньше или равна 150 2. Список клиентов из Москвы или из Самары 3. Список товаров, цена которых больше или равна 100 или меньше или равна 150 4. Список товаров, цена которых больше 100 и меньше 150	ОПК-6
11	4. Column1 AS MyColumn	Каким способом НЕЛЬЗЯ назначить псевдоним таблице или столбцу? 1. MyColumn = Column1 2. Column1 MyColumn 3. Column1 FOR MyColumn 4. Column1 AS MyColumn	ОПК-6