

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Заболотный Г.И.

Должность: Директор филиала

Дата подписания: 18.07.2025 10:57:13

Уникальный программный ключ:

476db7d4accb36ef8130172be235477473d63457266ce26b7e9e40f733b8b08

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Самарский государственный технический университет»

(ФГБОУ ВО «СамГТУ»)

УТВЕРЖДАЮ:

Директор филиала ФГБОУ ВО
"СамГТУ" в г. Новокуйбышевске

_____ / Г.И. Заболотни

" ____ " _____ 20 ____ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.05.02 «Языки и методы программирования»

**Код и направление подготовки
(специальность)**

09.03.01 Информатика и вычислительная
техника

Направленность (профиль)

Информатика и вычислительная техника в
нефтехимическом производстве

Квалификация

Бакалавр

Форма обучения

Очная

Год начала подготовки

2025

Институт / факультет

Кафедры филиала ФГБОУ ВО "СамГТУ" в г.
Новокуйбышевске

Выпускающая кафедра

кафедра "Информатика и системы
управления" (НФ-ИиСУ)

Кафедра-разработчик

кафедра "Информатика и системы
управления" (НФ-ИиСУ)

Объем дисциплины, ч. / з.е.

324 / 9

**Форма контроля (промежуточная
аттестация)**

Экзамен

Б1.О.05.02 «Языки и методы программирования»

Рабочая программа дисциплины разработана в соответствии с требованиями ФГОС ВО по направлению подготовки (специальности) **09.03.01 Информатика и вычислительная техника**, утвержденного приказом Министерства образования и науки РФ от № 929 от 19.09.2017 и соответствующего учебного плана.

Разработчик РПД:

Старший преподаватель

(должность, степень, ученое звание)

Т.П Рубцова

(ФИО)

Заведующий кафедрой

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

СОГЛАСОВАНО:

Председатель методического совета
факультета / института (или учебно-
методической комиссии)

А.А Малафеев, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

Руководитель образовательной
программы

А.В. Волкодаева, кандидат
экономических наук, доцент

(ФИО, степень, ученое звание)

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы	4
2. Место дисциплины (модуля) в структуре образовательной программы	4
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся	5
4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий	5
4.1 Содержание лекционных занятий	6
4.2 Содержание лабораторных занятий	8
4.3 Содержание практических занятий	9
4.4. Содержание самостоятельной работы	12
5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)	13
6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения	14
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем	14
8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)	15
9. Методические материалы	16
10. Фонд оценочных средств по дисциплине (модулю)	17

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

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
	ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками составления алгоритмов и программ, пригодных для практического применения
			Знать методы составления алгоритмов и программ, пригодных для практического применения
			Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
		ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса
			Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса
			Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса

2. Место дисциплины (модуля) в структуре образовательной программы

Место дисциплины (модуля) в структуре образовательной программы: **обязательная часть**

Код компетенции	Предшествующие дисциплины	Параллельно осваиваемые дисциплины	Последующие дисциплины
-----------------	---------------------------	------------------------------------	------------------------

ОПК-8			Выполнение и защита выпускной квалификационной работы; Технологии программирования
-------	--	--	---

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Вид учебной работы	Всего часов / часов в электронной форме	1 семестр часов / часов в электронной форме	2 семестр часов / часов в электронной форме
Аудиторная контактная работа (всего), в том числе:	128	48	80
Лекции	32	16	16
Практические занятия	80	32	48
Лабораторные работы	16	0	16
Самостоятельная работа (всего), в том числе:	124	60	64
подготовка к зачету	20	20	0
подготовка к лекциям	20	10	10
подготовка к практическим занятиям	54	30	24
подготовка к лабораторным работам	10	0	10
подготовка к экзамену	20	0	20
Контроль	72	36	36
Итого: час	324	144	180
Итого: з.е.	9	4	5

4. Содержание дисциплины (модуля), структурированное по темам (разделам), с указанием отведенного на них количества академических часов и видов учебных занятий

№ раздела	Наименование раздела дисциплины	Виды учебной нагрузки и их трудоемкость, часы				
		ЛЗ	ЛР	ПЗ	СРС	Всего часов
1	Развитие языков программирования. Парадигмы программирования	4	0	0	20	24
2	Основные элементы языков программирования	12	0	32	40	84
3	Методологии программирования	10	6	32	30	78
4	Методы проектирования и разработки программного обеспечения	6	10	16	34	66

	Контроль	0	0	0	0	72
	Итого	32	16	80	124	324

4.1 Содержание лекционных занятий

№ занятия	Наименование раздела	Тема лекции	Содержание лекции (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
1 семестр				
1	Развитие языков программирования. Парадигмы программирования	История и эволюция методов и языков программирования. Синтаксис, семантика, прагматика.	Эволюция методов и языков. Особенности формальных языков. Грамматики, классификация языков (контекстно-зависимые языки, контекстно-свободные языки, регулярные языки). Синтаксис и основные классы понятий, семантика и прагматика языков программирования.	2
2	Развитие языков программирования. Парадигмы программирования	Процедурная и структурная парадигмы. Объектно-ориентированная парадигма. Компонентно-ориентированное программирование. Событийная парадигма. Функциональная парадигма. Единство структурного-процедурного, объектного и функционального подходов. Реактивное программирование	Процедурная и структурная парадигмы. Программирование сверху-вниз. Программирование снизу-вверх. Низкоуровневые и высокоуровневые языки. Дихотомия императивного и декларативного подходов. Переход от императивного к декларативному программированию. Объектно-ориентированная парадигма. Функциональная парадигма. Основы парадигмальной декомпозиции. Средства поддержки автоматизации интеграционных решений и тестирования.	2
3	Основные элементы языков программирования	Основные элементы языков программирования	Введение в программирование на СИ. Структура программы. Подпрограммы, их назначение и классификация. Области доступности глобальных и локальных переменных. Объявление, инициализация.	2
4	Основные элементы языков программирования	Основные элементы языков программирования	Операции языка программирования Си. Массивы одномерные.	2
5	Основные элементы языков программирования	Стандартные структуры данных	Двумерные массивы.	2
6	Основные элементы языков программирования	Стандартные структуры данных	Программирование с использованием процедур и функций. Формальные и фактические параметры. Передача параметров различных типов по ссылке и значению. Рекурсия.	2
7	Основные элементы языков программирования	Стандартные структуры данных	Типы данных пользователя. Понятие абстрактного типа данных. Структурные типы данных. Записи. Списки, стеки, множества и объединения. Операции со структурными типами. Реализация структурных типов в языках программирования высокого уровня.	2
8	Основные элементы языков программирования	Стандартные структуры данных	Работа с файлами	2
Итого за семестр:				16
2 семестр				

9	Методологии программирования	Основные положения объектного подхода к разработке программ.	Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Способы распределения и заимствования шаблонов и поведения. Инкапсуляция и спецификация правил доступности элементов класса. Конструкторы и деструкторы. Наследование. Иерархия классов. Одиночное и множественное наследование. Способы реализации множественного наследования, их достоинства и недостатки. Таблицы виртуальных функций. Абстрактные базовые классы. Конечные производные классы.	2
10	Методологии программирования	Основные положения объектного подхода к разработке программ.	Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Способы распределения и заимствования шаблонов и поведения. Инкапсуляция и спецификация правил доступности элементов класса. Конструкторы и деструкторы. Наследование. Иерархия классов. Одиночное и множественное наследование. Способы реализации множественного наследования, их достоинства и недостатки. Таблицы виртуальных функций. Абстрактные базовые классы. Конечные производные классы.	2
11	Методологии программирования	Основные положения объектного подхода к разработке программ.	Принципы объектного подхода: абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость. Способы распределения и заимствования шаблонов и поведения. Инкапсуляция и спецификация правил доступности элементов класса. Конструкторы и деструкторы. Наследование. Иерархия классов. Одиночное и множественное наследование. Способы реализации множественного наследования, их достоинства и недостатки. Таблицы виртуальных функций. Абстрактные базовые классы. Конечные производные классы.	2

12	Методологии программирования	Отношения между объектами и/или классами	Полиморфизм. Ранее и позднее связывание. Управление последовательностью действий в объектно -ориентированной программе. Объект и процесс. Инициализация и взаимодействие объектов и процессов. Сообщения. Реализация механизмов посылки сообщений. Примеры функционирования объектно-ориентированной программы. Проектирование библиотек классов. Виды классов: конкретный тип, абстрактный тип, узловой класс, интерфейсный класс. Динамическая идентификация типа. Управление видимостью и областью действия имен. Управление памятью. Библиотеки контейнерных классов. Номенклатура контейнеров и примеры их использования. Иерархия классов исключений.	2
13	Методологии программирования	Отношения между объектами и/или классами	Полиморфизм. Ранее и позднее связывание. Управление последовательностью действий в объектно -ориентированной программе. Объект и процесс. Инициализация и взаимодействие объектов и процессов. Сообщения. Реализация механизмов посылки сообщений. Примеры функционирования объектно-ориентированной программы. Проектирование библиотек классов. Виды классов: конкретный тип, абстрактный тип, узловой класс, интерфейсный класс. Динамическая идентификация типа. Управление видимостью и областью действия имен. Управление памятью. Библиотеки контейнерных классов. Номенклатура контейнеров и примеры их использования. Иерархия классов исключений.	2
14	Методы проектирования и разработки программного обеспечения	Обзор методов проектирования и разработки программного обеспечения.	Основные подходы к разработке программ: процедурное, логическое, функциональное и объектно-ориентированное программирование. Процедурное и декларативное представление данных.	2
15	Методы проектирования и разработки программного обеспечения	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	Типы данных в языках программирования. Связь между данными и операциями. Абстрактные типы данных. Понятие интерфейса.	2
16	Методы проектирования и разработки программного обеспечения	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования	Типы данных в языках программирования. Связь между данными и операциями. Абстрактные типы данных. Понятие интерфейса.	2
Итого за семестр:				16
Итого:				32

4.2 Содержание лабораторных занятий

№ занятия	Наименование раздела	Тема лабораторного занятия	Содержание лабораторного занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
2 семестр				
1	Методологии программирования	Основы объектно-ориентированного программирования	Решение объектных задач.	2
2	Методологии программирования	Отношения между объектами и/или классами	Решение задач на разработку классов.	2
3	Методологии программирования	Отношения между объектами и/или классами	Решение задач на разработку классов.	2
4	Методы проектирования и разработки программного обеспечения	Основные подходы к разработке программ	Программирование статических изображений.	2
5	Методы проектирования и разработки программного обеспечения	Основные подходы к разработке программ	Программирование движущихся изображений.	2
6	Методы проектирования и разработки программного обеспечения	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования.	Решение задач на разработку интерфейсов.	2
7	Методы проектирования и разработки программного обеспечения	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования.	Решение задач на разработку интерфейсов.	2
8	Методы проектирования и разработки программного обеспечения	Связь структур данных и алгоритмов их обработки с управляющими структурами языков программирования.	Решение задач на разработку интерфейсов.	2
Итого за семестр:				16
Итого:				16

4.3 Содержание практических занятий

№ занятия	Наименование раздела	Тема практического занятия	Содержание практического занятия (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов / часов в электронной форме
1 семестр				
1	Основные элементы языков программирования	Стандартные структуры данных	Введение в программирование на СИ. Символы языка, структура программы. Переменные, константы. Объявление, инициализация.	2
2	Основные элементы языков программирования	Стандартные структуры данных	Стандартные типы данных в C++. Понятие операции. Преобразование типов.	2
3	Основные элементы языков программирования	Стандартные структуры данных	Линейные и разветвленные алгоритмы в C++. Операторы if и множественного выбор.	2
4	Основные элементы языков программирования	Стандартные структуры данных	Организация циклов в C++. Цикл с предусловием, цикл с постусловием.	2
5	Основные элементы языков программирования	Стандартные структуры данных	Организация циклов в C++. Цикл с параметром. Операторы break, continue, goto, exit.	2
6	Основные элементы языков программирования	Стандартные структуры данных	Функции в языке программирования C++. Объявление и определение функций, параметры функций. Локальные и глобальные переменные.	2
7	Основные элементы языков программирования	Стандартные структуры данных	Создание одномерных массивов в C++, особенности работы с ними	2
8	Основные элементы языков программирования	Стандартные структуры данных	Методы сортировки одномерных массивов в C++	2
9	Основные элементы языков программирования	Стандартные структуры данных	Создание многомерных массивов в C++, особенности работы с ними.	2
10	Основные элементы языков программирования	Стандартные структуры данных	Особенности сортировки матриц в C++	2
11	Основные элементы языков программирования	Стандартные структуры данных	Строка как массив символов. Ввод/вывод строк, функции работы со строками.	2
12	Основные элементы языков программирования	Стандартные структуры данных	Пользовательские типы данных в языке программирования C++. Структуры – особенности создания и работы.	2
13	Основные элементы языков программирования	Стандартные структуры данных	Пользовательские типы данных в языке программирования C++. Объединения и перечисления – особенности создания и работы.	2
14	Основные элементы языков программирования	Стандартные структуры данных	Организация ввода и вывода в языке программирования C++. Стандартные файлы ввода/вывода. Текстовые файлы. Методы обработки текстовых файлов.	2

15	Основные элементы языков программирования	Стандартные структуры данных	Организация ввода и вывода в языке программирования C++. Двоичные файлы. Последовательный доступ к элементам двоичных файлов. Произвольный доступ к элементам двоичных файлов.	2
16	Основные элементы языков программирования	Стандартные структуры данных	Функции в языке программирования C++. Строки, массивы и структуры в качестве параметров функций.	2
Итого за семестр:				32
2 семестр				
17	Методологии программирования	Стандартные структуры данных	Функции в языке программирования C++. Рекурсия. Встраиваемые функции. Перегрузка функций, использование аргументов по умолчанию.	2
18	Методологии программирования	Стандартные структуры данных	Указатели в языке программирования C++. Объявление и разыменовывание указателей.	2
19	Методологии программирования	Стандартные структуры данных	Указатели в языке программирования C++. Нулевые указатели и указатели на тип Void.	2
20	Методологии программирования	Стандартные структуры данных	Ссылки в языке программирования C++. Резервирование памяти в куче.	2
21	Методологии программирования	Модульное программирование	Модульное программирование	2
22	Методологии программирования	Модульное программирование	Модульное программирование	2
23	Методологии программирования	Основы объектно-ориентированного программирования	Основы ООП.	2
24	Методологии программирования	Основы объектно-ориентированного программирования	Основы ООП.	2
25	Методологии программирования	Основы объектно-ориентированного программирования	Наследование. Одиночное и множественное наследование. Способы реализации множественного наследования.	2
26	Методологии программирования	Основы объектно-ориентированного программирования	Наследование. Одиночное и множественное наследование. Способы реализации множественного наследования.	2
27	Методологии программирования	Основы объектно-ориентированного программирования	Наследование. Одиночное и множественное наследование. Способы реализации множественного наследования.	2
28	Методологии программирования	Основы объектно-ориентированного программирования	Проектирование библиотек классов.	2
29	Методологии программирования	Основы объектно-ориентированного программирования	Управление видимостью и областью действия имен. Управление памятью.	2

30	Методологии программирования	Основы объектно-ориентированного программирования	Библиотеки контейнерных классов. Номенклатура контейнеров и примеры их использования. Иерархия классов исключений.	2
31	Методологии программирования	Основы объектно-ориентированного программирования	Обработка исключений.	2
32	Методологии программирования	Основы объектно-ориентированного программирования	Примеры функционирования объектно-ориентированной программ.	2
33	Методы проектирования и разработки программного обеспечения	Методы программирования	Основные подходы к разработке программ.	2
34	Методы проектирования и разработки программного обеспечения	Методы программирования	Связь между данными и операциями.	2
35	Методы проектирования и разработки программного обеспечения	Методы программирования	Связь между данными и операциями.	2
36	Методы проектирования и разработки программного обеспечения	Паттерны проектирования	Паттерны проектирования	2
37	Методы проектирования и разработки программного обеспечения	Паттерны проектирования	Паттерны проектирования	2
38	Методы проектирования и разработки программного обеспечения	Паттерны проектирования	Паттерны проектирования	2
39	Методы проектирования и разработки программного обеспечения	Паттерны проектирования	Паттерны проектирования	2
40	Методы проектирования и разработки программного обеспечения	Паттерны проектирования	Паттерны проектирования	2
Итого за семестр:				48
Итого:				80

4.4. Содержание самостоятельной работы

Наименование раздела	Вид самостоятельной работы	Содержание самостоятельной работы (перечень дидактических единиц: рассматриваемых подтем, вопросов)	Количество часов
1 семестр			
Развитие языков программирования. Парадигмы программирования	подготовка к зачету	Развитие языков программирования. Парадигмы программирования.	20
Основные элементы языков программирования	подготовка к практическим занятиям	Основные элементы языка программирования C++.	20
Основные элементы языков программирования	подготовка к зачету	Основные элементы языка программирования C++.	20
Итого за семестр:			60
2 семестр			
Методологии программирования	подготовка к практическим занятиям	Основы объектно- ориентированного программирования.	30
Методы проектирования и разработки программного обеспечения	подготовка к экзамену	Методы программирования и основные элементы языка программирования C++.	34
Итого за семестр:			64
Итого:			124

5. Перечень учебной литературы и учебно-методического обеспечения по дисциплине (модулю)

№ п/п	Библиографическое описание	Ресурс НТБ СамГТУ (ЭБС СамГТУ, IPRbooks и т.д.)
Основная литература		
1	Алгоритмизация и программирование; Вузовское образование, 2019. - Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 80539	Электронный ресурс
2	Введение в программирование; Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 97539	Электронный ресурс
3	Программирование: основы языка C++; Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 102464	Электронный ресурс
Дополнительная литература		

4	Программирование в примерах и задачах; Лаборатория знаний, 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 99863	Электронный ресурс
5	Программирование на языке высокого уровня C/C++; МИСИ-МГСУ, ЭБС АСВ, 2018.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 76390	Электронный ресурс
6	Программирование: типовые задачи, алгоритмы, методы; Лаборатория знаний, 2020.- Режим доступа: https://elib.samgtu.ru/getinfo?uid=els_samgtu iprbooks 12264	Электронный ресурс

Доступ обучающихся к ЭР НТБ СамГТУ (elib.samgtu.ru) осуществляется посредством электронной информационной образовательной среды университета и сайта НТБ СамГТУ по логину и паролю.

6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения

При проведении лекционных занятий используется мультимедийное оборудование.

Организовано взаимодействие обучающегося и преподавателя с использованием электронной информационной образовательной среды университета.

№ п/п	Наименование	Производитель	Способ распространения
1	Microsoft Windows 8.1 Professional операционная система	Microsoft (Зарубежный)	Лицензионное
2	Microsoft Office 2013	Microsoft (Зарубежный)	Лицензионное
3	Dev C++ (версия 4.9.9.2, свободное ПО)	Компания Bloodshed Software (Зарубежный)	Свободно распространяемое

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», профессиональных баз данных, информационно-справочных систем

№ п/п	Наименование	Краткое описание	Режим доступа
1	eLIBRARY.ru	http://www.eLIBRARY.ru/	Российские базы данных ограниченного доступа
2	КонсультантПлюс (правовые документы) - доступ с ПК в Медиацентре (ауд. 42)	http://www.consultant.ru/	Российские базы данных ограниченного доступа
3	Электронно-библиотечная система IPRbooks	http://www.iprbookshop.ru/	Российские базы данных ограниченного доступа
4	Электронная библиотека изданий СамГТУ	http://irbis.samgtu.local/cgi-bin/irbis64r_01/cgiirbis_64.exe	Российские базы данных ограниченного доступа

8. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия

Аудитория № 302

Учебная аудитория для проведения занятий лекционного типа.

Помещение оснащено:

проектор, моноблок, экран;

имеется выход в сеть Интернет; с доступом в электронную информационно образовательную среду СамГТУ;

учебная мебель: 22 стола, 44 стула; стол и стул для преподавателя, кафедра, доска аудиторная

Практические занятия

Аудитория № 102

Аудитория для практических занятий и лабораторных работ, текущего контроля и промежуточной аттестации (для инвалидов и лиц ОВЗ)

Помещение оснащено:

компьютер в комплекте 8 шт: монитор;

Компьютер в комплекте 14 шт: монитор, сетевой фильтр;

имеется выход в сеть Интернет; и с доступом в электронную информационно образовательную среду СамГТУ;

учебная мебель: 23 компьютерных столов, 23 кресла-комфорт, 6 ученических парт, 12 ученических стульев, стол и стул преподавателя

Лабораторные занятия

Аудитория № 102

Аудитория для практических занятий и лабораторных работ, текущего контроля и промежуточной аттестации (для инвалидов и лиц ОВЗ)

Помещение оснащено:

компьютер в комплекте 8 шт: монитор;

Компьютер в комплекте 14 шт: монитор, сетевой фильтр;

имеется выход в сеть Интернет; и с доступом в электронную информационно образовательную среду СамГТУ;

учебная мебель: 23 компьютерных столов, 23 кресла-комфорт, 6 ученических парт, 12 ученических стульев, стол и стул преподавателя

Самостоятельная работа

Аудитория № 212

Учебная аудитория для проведения курсового проектирования групповых и индивидуальных консультаций и самостоятельной работы обучающихся

Помещение оснащено:

при необходимости используют ноутбук 4 шт.

имеется выход в сеть Интернет; с доступом в электронную информационно образовательную среду СамГТУ;

специализированная мебель: 4 ученических стола (2 пос. места), 8 ученических

стульев, стол и стул для преподавателя.

Аудитория № 304

Учебная аудитория для самостоятельной работы обучающихся.

Помещение оснащено:

при необходимости используют ноутбук 4 шт,

имеется выход в сеть Интернет; с доступом в электронную информационно образовательную среду СамГТУ;

Учебная мебель: 8 столов, 16 стульев, стол и стул для преподавателя

9. Методические материалы

Методические рекомендации при работе на лекции

До лекции студент должен просмотреть учебно-методическую и научную литературу по теме лекции с тем, чтобы иметь представление о проблемах, которые будут разбираться в лекции.

Перед началом лекции обучающимся сообщается тема лекции, план, вопросы, подлежащие рассмотрению, доводятся основные литературные источники. Весь учебный материал, сообщаемый преподавателем, должен не просто прослушиваться. Он должен быть активно воспринят, т.е. услышан, осмыслен, понят, зафиксирован на бумаге и закреплён в памяти. Приступая к слушанию нового учебного материала, полезно мысленно установить его связь с ранее изученным. Следя за техникой чтения лекции (акцент на существенном, повышение тона, изменение ритма, пауза и т.п.), необходимо вслед за преподавателем уметь выделять основные категории, законы и определять их содержание, проблемы, предполагать их возможные решения, доказательства и выводы. Осуществляя такую работу, можно значительно облегчить себе понимание учебного материала, его конспектирование и дальнейшее изучение.

Конспектирование лекции позволяет обработать, систематизировать и лучше сохранить полученную информацию с тем, чтобы в будущем можно было восстановить в памяти основные, содержательные моменты. Типичная ошибка, совершаемая обучающимся, дословное конспектирование речи преподавателя. Как правило, при записи «слово в слово» не остается времени на обдумывание, анализ и синтез информации. Отбирая нужную информацию, главные мысли, проблемы, решения и выводы, необходимо сокращать текст, строить его таким образом, чтобы потом можно было легко в нем разобраться. Желательно оставить в рабочих конспектах поля, на которых можно будет делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С окончанием лекции работа над конспектом не может считаться завершённой. Нужно еще восстановить отдельные места, проверить, все ли понятно, уточнить что-то на консультации и т.п. с тем, чтобы конспект мог быть использован в процессе подготовки к практическим занятиям, зачету, экзамену. Конспект лекции – незаменимый учебный документ, необходимый для самостоятельной работы.

Методические рекомендации при подготовке и работе на практическом занятии

Практические занятия по дисциплине проводятся в целях выработки практических умений и приобретения навыков в решении профессиональных задач.

Рекомендуется следующая схема подготовки к практическому занятию:

1. ознакомление с планом практического занятия, который отражает содержание предложенной темы;
2. проработка конспекта лекции;
3. чтение рекомендованной литературы;
4. подготовка ответов на вопросы плана практического занятия;
5. выполнение тестовых заданий, задач и др.

Подготовка обучающегося к практическому занятию производится по вопросам, разработанным

для каждой темы практических занятий и (или) лекций. В процессе подготовки к практическим занятиям, необходимо обратить особое внимание на самостоятельное изучение рекомендованной литературы.

Работа студентов во время практического занятия осуществляется на основе заданий, которые выдаются обучающимся в начале или во время занятия. На практических занятиях приветствуется активное участие в обсуждении конкретных ситуаций, способность на основе полученных знаний находить наиболее эффективные решения поставленных проблем, уметь находить полезный дополнительный материал по тематике занятий. Обучающимся необходимо обращать внимание на основные понятия, алгоритмы, определять практическую значимость рассматриваемых вопросов. На практических занятиях обучающиеся должны уметь выполнить расчет по заданным параметрам или выработать определенные решения по обозначенной проблеме. Задания могут быть групповые и индивидуальные. В зависимости от сложности предлагаемых заданий, целей занятия, общей подготовки обучающихся преподаватель может подсказать обучающимся алгоритм решения или первое действие, или указать общее направление рассуждений. Полученные результаты обсуждаются с позиций их адекватности или эффективности в рассмотренной ситуации.

Методические рекомендации по выполнению самостоятельной работы

Организация самостоятельной работы обучающихся ориентируется на активные методы овладения знаниями, развитие творческих способностей, переход от поточного к индивидуализированному обучению с учетом потребностей и возможностей обучающегося.

Самостоятельная работа с учебниками, учебными пособиями, научной, справочной литературой, материалами периодических изданий и Интернета является наиболее эффективным методом получения дополнительных знаний, позволяет значительно активизировать процесс овладения информацией, способствует более глубокому усвоению изучаемого материала. Все новые понятия по изучаемой теме необходимо выучить наизусть и внести в глоссарий, который целесообразно вести с самого начала изучения курса.

Самостоятельная работа реализуется:

- непосредственно в процессе аудиторных занятий;
- на лекциях, практических занятиях;
- в контакте с преподавателем вне рамок расписания;
- на консультациях по учебным вопросам, в ходе творческих контактов, при ликвидации задолженностей, при выполнении индивидуальных заданий и т.д.;
- в библиотеке, дома, на кафедре при выполнении обучающимся учебных и практических задач.

Эффективным средством осуществления обучающимся самостоятельной работы является электронная информационно-образовательная среда университета, которая обеспечивает доступ к учебным планам, рабочим программам дисциплин (модулей), практик, к изданиям электронных библиотечных систем.

10. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств представлен в приложении № 1.

**Фонд оценочных средств
по дисциплине
Б1.О.05.02 «Языки и методы программирования»**

Код и направление подготовки (специальность)	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Информатика и вычислительная техника в нефтехимическом производстве
Квалификация	Бакалавр
Форма обучения	Очная
Год начала подготовки	2025
Институт / факультет	Кафедры филиала ФГБОУ ВО "СамГТУ" в г. Новокуйбышевске
Выпускающая кафедра	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Кафедра-разработчик	кафедра "Информатика и системы управления" (НФ-ИиСУ)
Объем дисциплины, ч. / з.е.	324 / 9
Форма контроля (промежуточная аттестация)	Экзамен

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

Наименование категории (группы) компетенций	Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Результаты обучения (знать, уметь, владеть, соотнесенные с индикаторами достижения компетенции)
Общепрофессиональные компетенции			
	ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками составления алгоритмов и программ, пригодных для практического применения
			Знать методы составления алгоритмов и программ, пригодных для практического применения
			Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения
		ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса
			Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса
			Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса

Матрица соответствия оценочных средств запланированным результатам обучения

Код индикатора достижения компетенции	Результаты обучения	Оценочные средства	Текущий контроль успеваемости	Промежуточная аттестация
---------------------------------------	---------------------	--------------------	-------------------------------	--------------------------

Развитие языков программирования. Парадигмы программирования				
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Знать методы составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	практические задачи	Да	Нет
	Знать методы составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
Основные элементы языков программирования				
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Знать методы составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да

	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Знать методы составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	практические задачи	Да	Нет
ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
Методологии программирования				
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Знать методы составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Знать методы составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	практические задачи	Да	Нет

	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
Методы проектирования и разработки программного обеспечения				
ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования	Знать методы составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	оценочные средства промежуточного контроля	Нет	Да
	Знать методы составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Владеть навыками составления алгоритмов и программ, пригодных для практического применения	практические задачи	Да	Нет
	Уметь составлять алгоритмы, писать и отлаживать коды, пригодные для практического применения	практические задачи	Да	Нет

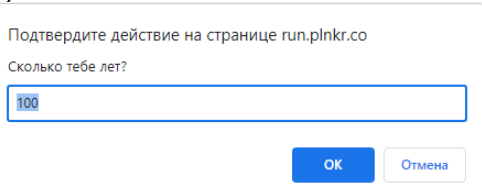
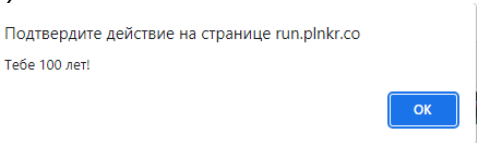
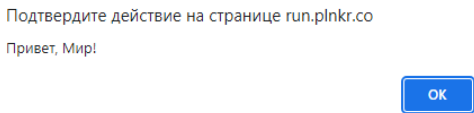
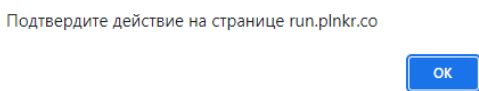
ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	практические задачи	Да	Нет
	Владеть навыками применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Уметь применять языки программирования, современные технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да
	Знать методы применения языков программирования, современных технологии для решения прикладных задач различного класса	оценочные средства промежуточного контроля	Нет	Да

ТЕСТОВЫЕ ЗАДАНИЯ

Направление подготовки 09.03.01 «Информатика и вычислительная техника»
Дисциплина: «Языки и методы программирования»

Перечень компетенций и индикаторов достижения компетенций, для оценки
сформированности которых используется данные ФОС

Код и наименование компетенции	Код и наименование индикатора достижения компетенции, реализуемые дисциплиной
ОПК-8 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1 Составляет алгоритмы, пишет и отлаживает коды на языке программирования
	ОПК-8.2 Применяет языки программирования, современные программные среды разработки информационных систем и технологии для решения прикладных задач различного класса

Номер задания	Содержание вопроса	Правильный ответ на задание
1.	Выберите правильный вариант ответа. По предложенному коду JavaScript определите первое окно отображения. <pre><!DOCTYPE html> <script> "use strict"; let age = prompt("Сколько тебе лет?", 100); alert(`Тебе \${age} лет!`); // Тебе 100 лет! </script></pre> A)  B)  C)  D) 	A
2.	Выберите правильный вариант ответа. Представлен код JavaScript. Результат работы программы при внесении возраста 18?	C

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre><!DOCTYPE html> <script> "use strict"; let age = prompt('Возраст?', 18); let message = (age < 3) ? 'Здравствуй, малыш!' : (age < 18) ? 'Привет!' : (age < 100) ? 'Здравствуйте!' : 'Какой необычный возраст!'; alert(message); </script></pre> A) Здравствуй, малыш! B) Привет! C) Здравствуйте! D) Какой необычный возраст!	
3.	Выберите правильный вариант ответа. Представлен код JavaScript. По данному коду определите шаг цикла. <pre><!DOCTYPE html> <script> "use strict"; let i = 0; for (i = 0; i < 3; i++) { // используем существующую переменную alert(i); // 0, 1, 2 } alert(i); // 3, переменная доступна, т.к. была объявлена снаружи цикла </script></pre> A) значение i при каждой итерации увеличивается на 1 B) значение i при каждой итерации увеличивается на 2 C) значение i при каждой итерации увеличивается на 3 D) значение i при каждой итерации увеличивается на 4	A
4.	Выберите правильный вариант ответа. Представлен код JavaScript. Проанализируйте алгоритм кода и определите тип используемой переменной b. <pre>let a = 7; function sum(a) { let b = 8; console.log(a + b); } sum(a); console.log(b);</pre> A) локальная переменная B) внешняя переменная C) аргументируемая переменная D) переменная модуля	A
5.	Выберите правильный вариант ответа. Представлен код JavaScript. Проанализируйте алгоритм кода и определите количество возможных вариантов ответа.	B

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> <!DOCTYPE html> <script> "use strict"; function checkAge(age) { if (age >= 18) { return true; } else { return confirm('А родители разрешили?'); } } let age = prompt('Сколько вам лет?', 18); if (checkAge(age)) { alert('Доступ получен'); } else { alert('Доступ закрыт'); } </script> </pre> A) 2 B) 3 C) 4 D) 6	
6.	Выберите правильный вариант ответа. Поставлена задача JavaScript. Нужно сохранить в переменную одно, либо другое значение, в зависимости от условия, то можно это сделать двумя путями. A) let value = 1 // значение по умолчанию if (day === 'Tuesday') { value = 50 } B) let value = 0 // значение по умолчанию if (externalValue) { value = externalValue // установить значение, если в externalValue что-либо хранится } C) const value = externalValue ?? 42 D) let value = 0 if (day === 'Tuesday') { value = 50 }	A
7.	Выберите правильный вариант ответа. Поставлена задача JavaScript. Установить значение по умолчанию, если в переменной нет значения A) let value = 1 // значение по умолчанию if (day === 'Tuesday') { value = 50 } B) let value = 0 // значение по умолчанию if (externalValue) { value = externalValue // установить значение, если в externalValue что-либо хранится }	B

Номер задания	Содержание вопроса	Правильный ответ на задание
	C) const value = externalValue ?? 42 D) let value = 0 if (day === 'Tuesday') { value = 50 }	
8.	Выберите правильный вариант ответа. В данной конструкции программного кода JavaScript значение переменной shoppingDone. <pre>var shoppingDone = false; if (shoppingDone === true) { var childsAllowance = 10; } else { var childsAllowance = 5; }</pre> A) shoppingDone всегда равный false B) shoppingDone меняется на false C) shoppingDone не меняется на false D) shoppingDone не меняется на true	A
9.	Выберите правильный вариант ответа. Проанализируйте алгоритм кода JavaScript и выберите правильный вариант отображения данных программы. <pre>var output = ""; for (var i = 1; i <= 9; i++) { for (var j = 1; j <= 9; j++) { output += ' ' + i * j; if (i * j < 10) { output += ' '; } } console.log(output); output = ""; }</pre> A) <pre>1 2 3 4 5 6 7 8 9 2 4 6 8 10 12 14 16 18 3 6 9 12 15 18 21 24</pre> B) <pre>1 2 3 4 5 6 7 8 9 2 4 6 8 10 12 14 16 18 3 6 9 12 15 18 21 24 27 4 8 12 16 20 24 28 32 36 5 10 15 20 25 30 35 40 45 6 12 18 24 30 36 42 48 54 7 14 21 28 35 42 49 56 63 8 16 24 32 40 48 56 64 72 9 18 27 36 45 54 63 72 81</pre> C) <pre>4 6 8 10 12 14 16 18 6 9 12 15 18 21 24 27 8 12 16 20 24 28 32 36 10 15 20 25 30 35 40 45 12 18 24 30 36 42 48 54 14 21 28 35 42 49 56 63 16 24 32 40 48 56 64 72 18 27 36 45 54 63 72 81</pre> D) <pre>4 6 8 10 6 9 12 15 8 12 16 20 10 15 20 25 12 18 24 30 14 21 28 35 16 24 32 40 18 27 36 45</pre>	B
10.	Выберите правильный вариант ответа.	C

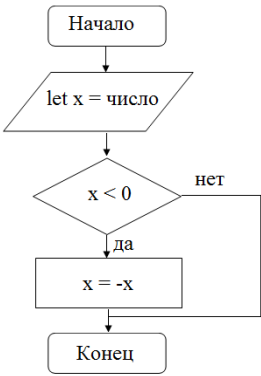
Номер задания	Содержание вопроса	Правильный ответ на задание
	Рассмотрите алгоритм предложенного кода и определите количество функций в коде. <pre>const factorial = (n) => { if (n === 0) { return 1; } const iter = (counter, acc) => { if (counter === 1) { return acc; } return iter(counter - 1, counter * acc); }; return iter(n, 1); };</pre> A) 0 B) 1 C) 2 D) 4	
11.	Выберите правильный вариант ответа. Рассмотрите предложенный код и определите характеристику вычисления. <pre>const factorial = (n) => { if (n === 0) { return 1; } const iter = (counter, acc) => { if (counter === 1) { return acc; } return iter(counter - 1, counter * acc); }; return iter(n, 1); };</pre> A) Итеративный алгоритм вычисления B) Рекурсивный алгоритм вычисления C) Пассивный алгоритм вычисления D) Иерархический алгоритм вычисления	B
12.	Выберите правильный вариант ответа. Представлен код PHP. Определите отображение данных работы кода. <pre>\$i = 0; for (; \$i < 10;) { echo \$i; \$i += 1; }</pre> A) 0123456789 B) 02468 C) 0246810 D) 02468	A
13.	Выберите правильный вариант ответа. Представлен код PHP. Определите количество выполненных итераций <pre>for (\$i=1, \$sum=-1; false, true, false; \$i = \$i + 3, \$sum = \$sum+3) {</pre>	A

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>print "\$i "; print "\$sum< br >"; }</pre> A) 0 B) 1 C) 2 D) 4	
14.	Выберите правильный вариант ответа. Представлен код PHP. Определите индекс последнего массива <pre>\$parts[2] = "K155ЛАЗ"; \$parts[7] = "K155ЛА8"; \$parts[20] = "K155ЛН1"; \$parts[] = "КРЕН5В";</pre> A) 20 B) 21 C) 22 D) 23	В
15.	Выберите правильный вариант ответа. Для чего используется цикл foreach. A) цикл foreach используется для работы со списками B) цикл foreach используется для работы с переменными C) цикл foreach используется для работы с массивом D) цикл foreach используется для работы с объектами	С
16.	По предложенному коду JavaScript опишите алгоритм изменения переменной \$. <pre><!DOCTYPE html> <script> "use strict"; let name = "Иван"; alert(`Привет, \${name}!`); alert(`результат: \${1 + 2}`); </script></pre>	Переменная \$ в ходе программы изменялась: Иван 3
17.	По предложенному коду JavaScript опишите алгоритм изменения переменной \$. <pre><!DOCTYPE html> <script> "use strict"; let name = "Иван"; alert(`Привет, \${name}!`); alert(`результат: \${1 + 2}`); </script></pre>	Переменная \$ в ходе программы изменялась: 1 Name Ilya
18.	По предложенному коду JavaScript опишите алгоритм изменения переменной Number. <pre><!DOCTYPE html> <script> "use strict"; alert(Number(" 123 ")); alert(Number(false)); alert(Number(true)); </script></pre>	Переменная Number в ходе программы изменялась: 123 0 1
19.	По предложенному коду JavaScript опишите алгоритм отображения данных в модальных окнах. <pre><!DOCTYPE html></pre>	В модальных окнах отображаются следующие данные: true

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre><script> "use strict"; alert(Boolean(1)); alert(Boolean(0)); alert(Boolean("Привет!")); alert(Boolean("")); // </script></pre>	false true false
20.	По предложенному коду JavaScript опишите какие действия применены к переменной x. <pre><!DOCTYPE html> <script> "use strict"; let x = 1; x = -x; alert(x); </script></pre>	К значению переменной x применен унарный минус. Значение x стало -1.
21.	Представлен код JavaScript. Перечислите значения, отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(5 % 2); alert(8 % 3); alert(8 % 4); </script></pre>	В модальных окнах будут отображаться значения остатков от деления: 1 2 0
22.	Представлен код JavaScript. Перечислите значения, отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(2 ** 2); alert(2 ** 3); alert(2 ** 4); </script></pre>	В модальных окнах будут отображаться значения возведения в степень 2: 4 8 16
23.	Представлен код JavaScript. Перечислите значения, отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(4 ** (1/2)); alert(8 ** (1/3)); </script></pre>	В модальных окнах будут отображаться значения взятию корня и куба 2: 2 2
24.	Представлен код JavaScript. Перечислите значения, отображаемые в модальных окнах? <pre><!DOCTYPE html> <script> "use strict"; alert(2 > 1); alert(2 == 1); alert(2 != 1); </script></pre>	В модальных окнах будут отображаться результаты работы операторов сравнения: true false true
25.	Представлен код JavaScript. По алгоритму данного кода определите действия после знака «?».	В зависимости от условия compare ==

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre> <!DOCTYPE html> <script> "use strict"; let company = prompt('Какая компания создала JavaScript?', ''); (company == 'Netscape') ? alert('Верно!') : alert('Неправильно.');</pre>	'Netscape', будет выполнена либо первая, либо вторая часть после?.
26.	Представлен код JavaScript. Кратко опишите действия директивы break в данной программе. <pre> <!DOCTYPE html> <script> "use strict"; let sum = 0; while (true) { let value = +prompt("Введите число", ""); if (!value) break; sum += value; } alert('Сумма: ' + sum);</pre>	Директива break полностью прекращает выполнение цикла и передаёт управление на строку за его телом, то есть на alert.
27.	Представлен код JavaScript. Кратко опишите действия директивы continue в данной программе. <pre> <!DOCTYPE html> <script> "use strict"; for (let i = 0; i < 10; i++) { if (i % 2 == 0) continue; alert(i); // 1, затем 3, 5, 7, 9 }</pre>	Для чётных значений i, директива continue прекращает выполнение тела цикла и передаёт управление на следующую итерацию for (со следующим числом). alert вызывается только для нечётных значений.
28.	Представлен пример конструкции <pre> const number = prompt("Введите число: "); if (number >= 0) { if (number == 0) { console.log("Это ноль."); } else { console.log("Это положительное число."); } } else { console.log("Это отрицательное число."); }</pre>	Вложенные операторы if...else.
29.	Представлен код JavaScript. Перечислите аргументы объявленной функции при ее запуске. <pre> function sayWelcome (userFirstName, userLastName) { console.log(`Добро пожаловать, \${userLastName} \${userFirstName}`); } sayWelcome('Иван', 'Иванов'); sayWelcome('Петр', 'Петров'); sayWelcome('Дмитрий');</pre>	Для функции sayWelcome в данном коде использовались аргументы: Иван, Иванов Петр, Петров Дмитрий
30.	Представлен код JavaScript. Проанализируйте алгоритм кода и опишите кратко действие данного кода.	Представлен код проверка: принимает ли некая

Номер задания	Содержание вопроса	Правильный ответ на задание
	<pre>let item = 3; switch(item) { case 1: console.log("item = 1"); case 2: console.log("item = 2"); case 3: console.log("item = 3"); case 4: console.log("item = 4"); default: console.log("item другое значение"); }</pre>	переменная одно из значений 1, 2, 3 или 4.
31.	Представлен код JavaScript. Проанализируйте алгоритм кода и опишите работу строки <code>user.firstname = 'Александр';</code>. <pre>const someUser = { firstname: 'Петр', lastname: 'Петров' } function changeUserName(user) { user.firstname = 'Александр'; } changeUserName(someUser); console.log(someUser.firstname);</pre>	Объявлена переменная <code>someUser</code> и присвоен ей объект, состоящий из двух свойств. Строкой <code>user.firstname = 'Александр';</code> меняем значение свойства <code>firstname</code> на новое. В итоге при вызове функции свойство <code>firstname</code> будет 'Александр'.
32.	Опишите понятие Колбэк функция (от английского callback function) в JavaScript.	Колбэк функция- обычная функция, которая просто вызывается внутри другой функции. Такие функции ещё называются функциями обратного вызова. Они очень часто применяются в асинхронном коде.
33.	Опишите понятие Рекурсия в JavaScript.	Рекурсия возможность вызова функцией внутри самой себя.
34.	Представлен код страницы HTML с скриптом на JavaScript. Кратко опишите действия скрипта. Файл HTML <pre><div class="password"> <input type="password" id="password-input" placeholder="Введите пароль" name="password"> <label><input type="checkbox" class="password- checkbox"> Показать пароль</label> </div></pre> Файл JavaScript. <pre>\$('body').on('click', '.password-checkbox', function() { if (\$(this).is(':checked')) { \$('#password-input').attr('type', 'text'); } else { \$('#password-input').attr('type', 'password'); } });</pre>	Показывает пароль по чекбоксу.
35.	Опишите кратко алгоритм, представленный на рисунке.	Вычисление модуля числа, хранящегося в переменной <code>x</code> .

Номер задания	Содержание вопроса	Правильный ответ на задание
	 <pre> graph TD Start([Начало]) --> Assign[/let x = число/] Assign --> Decision{x < 0} Decision -- да --> Negate[x = -x] Decision -- нет --> End([Конец]) Negate --> End </pre>	
36.	Проанализируйте алгоритм программного кода JavaScript. Кратко опишите действия кода. <pre> let x = 4; if(x >= 2 && x <= 7) console.log("x попадает в [2; 7]"); else console.log("x не попадает в [2; 7]"); </pre>	Две проверки: первая – проверяем, что $x \geq 2$ и вторая – проверяем, что $x \leq 7$. Для проверки используется логический оператор И (&&) (логическая конъюнкция). Если оба этих условия выполняются одновременно, то x попадает в диапазон [2; 7].
37.	Проанализируйте алгоритм программного кода JavaScript. Кратко опишите действия кода. <pre> let x = 40; if(x < 2 x > 7) console.log("x не попадает в [2; 7]"); else console.log("x попадает в [2; 7]"); </pre>	В составном условии используется связка по ИЛИ (две вертикальные черты – это ИЛИ) и оно будет истинно, если истинно или первое, или второе условие. То есть, в нашем случае, если $x < 2$ или $x > 7$, то делается вывод о невыхождении переменной x в указанный диапазон.
38.	Представлен алгоритм конструкции цикла for в JavaScript. По предложенному коду определите инициацию.  <pre> for (var i = 1; i <= 8; i++) { console.log(i); } </pre>	Инициализация - это выражение, которое выполняется один раз перед выполнением цикла; обычно используется для инициализации счётчика. Инициализация в предложенном коде: <code>var i = 1</code> (объявление переменной i и присвоение ей значения 1).
39.	Представлен алгоритм конструкции цикла for в JavaScript. По предложенному коду определите условие.	Условие - это выражение, истинность которого проверяется перед каждой итерацией; если выражение вычисляется как истина, то выполняется итерация; в противном случае цикл «for» завершает работу. Условие выполнения цикла в данном коде: $i \leq 8$ (пока значение переменной i меньше или

Номер задания	Содержание вопроса	Правильный ответ на задание
	 <pre>for (var i = 1; i <= 8; i++) { console.log(i); }</pre>	равно 8).
40.	Представлен алгоритм конструкции цикла for в JavaScript. По предложенному коду определите инструкцию.  <pre>for (var i = 1; i <= 8; i++) { console.log(i); }</pre>	Инструкция в предложенном коде, которую нужно выполнять: console.log(i) (выведение значения счётчика в консоль).
41.	Представлен алгоритм конструкции цикла while в JavaScript. По предложенному коду определите условие.  <pre>let a = 0; while (a <= 8) { a++; if (a % 2 !== 0) { continue; } console.log(a); }</pre>	Цикл while с условием a <= 8
42.	Представлен алгоритм конструкции цикла while в JavaScript. По предложенному коду опишите кратко инструкции кода.	Инструкции прописаны в фигурных скобках: 1) увеличим значение переменной a на 1 2) если число нечётное (остаток от деления на 2 не равен 0), то... 3) пропустим дальнейшее выполнение текущей итерации и перейдём к

Номер задания	Содержание вопроса	Правильный ответ на задание
	 <pre> let a = 0; while (a <= 8) { a++; if (a % 2 !== 0) { continue; } console.log(a); } </pre>	следующей 4) выведем значение переменной a в консоль
43.	Представлен алгоритм конструкции цикла do...while в JavaScript. По предложенному коду определите инструкции (тело цикла).  <pre> let num, sum = 0; do { num = +prompt ('Введите число', ''); if (num) { sum += num; } } while (num); console.log(sum); </pre>	Инструкции прописаны в фигурных скобках между операторами do и while.
44.	Проанализируйте алгоритм программного кода JavaScript. Что из себя представляет someLabel. <pre> someLabel: while (условие) { if (условие) { break someLabel; } } </pre>	someLabel является меткой. Вызов break someLabel приведёт к переходу в конец цикла, перед которым данная метка указана.
45.	Представлен код JavaScript. Кратко опишите действия данного кода с точки зрения использования метки. <pre> passLoop: for (n = 0; n < 4; n++) { for (i = 0; i < 4; i++) { if (n === 2 && i === 2) { continue passLoop; } console.log('n = ' + n + ', i = ' + i); } } </pre>	При достижении условия (n === 2 && i === 2) команда continue passLoop завершает текущую итерацию и переносит поток выполнения кода в начало верхнего цикла.

Номер задания	Содержание вопроса	Правильный ответ на задание
46.	Представлен код на JavaScript. Какая из функций используется как колбэк . <pre>function printToLog(message) { console.log(message); } function sum(num1, num2, callback) { const result = num1 + num2; callback(result); } sum(5, 11, printToLog);</pre>	printToLog – её будем использовать в роли колбэк функции, т.е. вызывать в sum
47.	Поясните понятие «динамическая типизация» применяемая в PHP <pre><?php \$id = 123; echo "<p>id = \$id</p>"; \$id = "jhveruuyeru"; echo "<p>id = \$id</p>"; ?></pre>	PHP - язык с динамической типизацией, мы можем присваивать одной и той же переменной значения разных типов в процессе выполнения кода.
48.	Представлен код PHP. Результат выполнения кода при <pre>a=0. \$a = 5; if(\$a>0){ echo "Переменная а больше нуля"; } elseif(\$a < 0){ echo "Переменная а меньше нуля"; } else{ echo "Переменная а равна нулю"; }</pre>	В браузере будет отображен текст: Переменная а равна нулю
49.	Дайте характеристику применения второго кода PHP на приведенной странице WEB. <pre><!DOCTYPE html> <html> <head> <title>METANIT.COM</title> <meta charset="utf-8" /> </head> <body> <?php \$a = 5; ?> <?php if (\$a > 0) { ?> <h2>Переменная а больше нуля</h2> <?php } ?> </body> </html></pre>	Можем переключаться внутри конструкции на код HTML. В данном случае само условие указывается в отдельном блоке php: <?php if (\$a > 0) { ?>. Важно, что при этом этот блок содержит только открывающую фигурную скобку "{". Завершается конструкция if другим блоком php, который содержит закрывающую фигурную скобку: <?php } ?>
50.	Представлен код PHP. Изучите алгоритм кода. Кратко опишите действие кода. <pre>\$user = array("name" => "Sergei Loftoff", "email" => "loftoff@gmail.com", "age" => 23); //Получение доступа к значениям в array echo "Name: " . \$user["name"] . "\n"; echo "Email: " . \$user["email"] . "\n"; echo "Age: " . \$user["age"] . "\n";</pre>	В этом примере мы создали ассоциативный массив \$user, в котором хранится информация о пользователе. Мы можем получить доступ к значениям, хранящимся в нем, используя ключи (например, "name", "email", "age"), указанные в квадратных скобках после имени массива.

Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие процессы формирования компетенций

Характеристика процедуры текущего контроля успеваемости и промежуточной аттестации по дисциплине

Оценивание знаний, умений, навыков и опыта деятельности проводятся на основе сведений, приводимых в матрице соответствия оценочных средств запланированным результатам обучения.

Цель текущего контроля успеваемости и промежуточной аттестации по учебным дисциплинам в семестре – проверка приобретаемых обучающимися знаний, умений, навыков в контексте формирования установленных образовательной программой компетенций в течение семестра.

Шкала оценивания:

«Отлично» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки «неудовлетворительно»: студент показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи повышенной сложности, свободно использовать справочную литературу, делать обоснованные выводы из результатов анализа конкретных ситуаций;

«Хорошо» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «хорошо» и «отлично», при условии отсутствия оценки

«неудовлетворительно», допускается оценка «удовлетворительно»: обучающийся показал прочные знания основных положений фактического материала, умение самостоятельно решать конкретные практические задачи, предусмотренные рабочей программой, ориентироваться в рекомендованной справочной литературе, умеет правильно оценить полученные результаты анализа конкретных ситуаций;

«Удовлетворительно» – выставляется, если сформированность заявленных образовательных результатов компетенций оценивается критериями «удовлетворительно», «хорошо» и «отлично»: обучающийся показал знание основных положений фактического материала, умение получить с помощью преподавателя правильное решение конкретной практической задачи из числа предусмотренных рабочей программой, знакомство с рекомендованной справочной литературой;

«Неудовлетворительно» – выставляется, если при ответе обучающегося выявились существенные пробелы в знаниях основных положений фактического материала, неумение с помощью преподавателя получить правильное решение конкретной практической задачи из числа предусмотренных рабочей программой учебной дисциплины.

Ответы и решения, обучающихся оцениваются по следующим общим критериям: распознавание проблем; определение значимой информации; анализ проблем; аргументированность; использование стратегий; творческий подход; выводы; общая грамотность.

Обучающиеся обязаны сдавать все задания в сроки, установленные преподавателем. Оценка

«Удовлетворительно» по дисциплине, может выставляться и при неполной сформированности компетенций в ходе освоения отдельной учебной дисциплины, если их формирование предполагается продолжить на более поздних этапах обучения, в ходе изучения других учебных дисциплин.

Текущий контроль осуществляется через систему оценки преподавателем всех видов работ обучающихся, предусмотренных рабочей программой дисциплины и учебным планом.

Критерии оценки теста.

Количество верных ответов:

80-100% -оценка «отлично»: обучающийся демонстрирует глубокое знание учебно-программного материала, умение свободно выполнять задания, усвоивший взаимосвязь основных понятий дисциплины; способный самостоятельно приобретать новые знания и умения; способный самостоятельно использовать углубленные знания;

71-85% -оценка «хорошо»: обучающийся демонстрирует полное знание учебно-программного материала, успешно выполняющий предусмотренные программой задания, показывающий систематический характер знаний по дисциплине и способный к их самостоятельному пополнению и обновлению в ходе дальнейшего обучения в вузе и в будущей профессиональной деятельности;

50-70% -оценка «удовлетворительно»: обучающийся обнаруживает знание основного учебного программного материала в объеме, необходимом для дальнейшего обучения, выполняющего задания, предусмотренные программой, допустившим неточности в ответе, но обладающим необходимыми знаниями для их устранения;

менее 50% -оценка «неудовлетворительно»: обучающийся демонстрирует пробелы в знаниях основного учебного программного материала, допускает принципиальные ошибки в выполнении предусмотренных программой заданий.

На этапе промежуточной аттестации используется система оценки успеваемости обучающихся, которая позволяет преподавателю оценить сформированность планируемых результатов обучения, а также уровень освоения материала обучающимися.

Форма оценки знаний: оценка - 5 «отлично»; 4 «хорошо»; 3 «удовлетворительно»; 2 «неудовлетворительно». возможно использовать балльно-рейтинговые оценки.

Основанием для определения оценки на зачете служит уровень освоения обучающимся материала и формирования компетенция, предусмотренных учебным планом.

Успеваемость на зачете определяется оценками: «зачтено»; «не зачтено».

Оценка	Критерии оценивания	Балльно-рейтинговая оценка
«Зачтено»	Обучающийся освоил компетенции дисциплины на 51-100 % и показал хорошие знания изученного учебного материала, логично и последовательно изложил и полностью раскрыл смысл предлагаемого вопроса; продемонстрировал умение применить теоретические знания для решения практической задачи; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	51-100
«Не зачтено»	Обучающийся освоил компетенции дисциплины менее чем на 51% и при ответе на предлагаемый вопрос выявились существенные пробелы в знаниях учебного материала, неумение с помощью преподавателя получить правильное решение практической задачи; не в полном объеме выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	0- 50

Основанием для определения оценки на экзамене служит уровень освоения обучающимся учебного материала, умение решать практические задачи и формирования компетенция, предусмотренных учебным планом.

Успеваемость на экзамене определяется оценками: «отлично»; «хорошо»; «удовлетворительно»; «не удовлетворительно».

Оценка	Критерии оценивания	Балльно-рейтинговая оценка
«Отлично»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 86-100 %, показал глубокие знания учебного материала, логично и последовательно изложил содержание ответов на вопросы билета; продемонстрировал умение иллюстрировать теоретические положения конкретными примерами и свободно выполнять экзаменационные задания; усвоил основную и ознакомился с дополнительной литературой; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	86-100
«Хорошо»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 61-85 %, показал глубокие знания учебного материала, логично и последовательно изложил содержание ответов на вопросы билета, но допустил несущественные неточности; продемонстрировал умение иллюстрировать теоретические положения конкретными примерами и выполнять экзаменационные задания; усвоил основную и ознакомился с дополнительной литературой; выполнил все контрольные задания, предусмотренные рабочей программой дисциплины	61-85
«Удовлетворительно»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования на 51-60 %, показал знания учебного материала в объеме, необходимом для дальнейшего освоения учебных программ, но допустил погрешности в изложении ответов на вопросы билета и при выполнении экзаменационных заданий; ознакомился с основной литературой, рекомендованной программой; справился с контрольными заданиями, предусмотренными рабочей программой дисциплины	51-60
«Не удовлетворительно»	Обучающийся освоил компетенции дисциплины на всех этапах их формирования менее чем на 51 %, обнаружил пробелы в знаниях учебного материала, допустил принципиальные ошибки в	0-50

	выполнении контрольных заданий, предусмотренных рабочей программой дисциплины	
--	---	--

Интегральная оценка

Критерии	Традиционная оценка	Балльно-рейтинговая оценка
5	5	86 - 100
4	4	61-85
3	3	51-60
2 и 1	2, Незачет	0-50
5, 4, 3	Зачет	51-100