МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Проректор по учебной работе
_____П.В. Сенченко
«23» _____12_____ 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ИНФОРМАТИКА И ПРОГРАММИРОВАНИЕ

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **38.03.05 Бизнес-информатика**
Направленность (профиль) / специализация: **ИТ-предпринимательство**
Форма обучения: **заочная (в том числе с применением дистанционных образовательных технологий)**
Факультет: **Факультет дистанционного обучения (ФДО)**
Кафедра: **Кафедра автоматизации обработки информации (АОИ)**
Курс: **1, 2**
Семестр: **1, 2, 3**
Учебный план набора 2021 года

Объем дисциплины и виды учебной деятельности

Виды учебной деятельности	1 семестр	2 семестр	3 семестр	Всего	Единицы
Лабораторные занятия	8	8	12	28	часов
Самостоятельная работа	49	49	79	177	часов
Самостоятельная работа под руководством преподавателя	4	4	6	14	часов
Контрольные работы	2	2	2	6	часов
Подготовка и сдача экзамена	9	9	9	27	часов
Общая трудоемкость (включая промежуточную аттестацию)	72	72	108	252 7	часов з.е.

Формы промежуточной аттестация	Семестр	Количество
Экзамен	1	
Контрольные работы	1	1
Экзамен	2	
Контрольные работы	2	1
Экзамен	3	
Контрольные работы	3	1

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Сенченко П.В.
Должность: Проректор по УР
Дата подписания: 23.12.2020
Уникальный программный ключ:
a1119608-cdff-4455-b54e-5235117c185c

Томск

Согласована на портале № 65300

1. Общие положения

1.1. Цели дисциплины

1. Развитие теоретических представлений и практических навыков работы с информацией, хранящейся или обрабатываемой в вычислительных системах; формирование навыков работы с компьютером как средством управления информацией; формирование навыков работы с информацией из различных источников, в том числе в глобальных компьютерных сетях; формирование навыков алгоритмизации и структурного программирования; формирование навыков объектно-ориентированного мышления.

1.2. Задачи дисциплины

1. Формирование у студента знаний основных понятий, концепций, принципов и теорий, связанных с информатикой, понятия количества информации, типов систем счисления, основных принципов структурного программирования; обучение студентов навыкам графического представления алгоритмов; обучение студентов навыкам разработки и отладки программ на алгоритмических языках программирования; изучение техники объектно-ориентированного анализа; изучение приемов объектно-ориентированного программирования.

2. Место дисциплины в структуре ОПОП

Блок дисциплин: Б1. Дисциплины (модули).
Часть блока дисциплин: Обязательная часть.
Модуль дисциплин: Модуль направления подготовки (special hard skills – SHS).
Индекс дисциплины: Б1.О.03.01.
Реализуется с применением электронного обучения, дистанционных образовательных технологий.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО и основной образовательной программой (таблица 3.1):

Таблица 3.1 – Компетенции и индикаторы их достижения

Компетенция	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине
Универсальные компетенции		
-	-	-
Общепрофессиональные компетенции		

Согласована на портале № 65300

ОПК-3. Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3.1. Знает теоретические основы процессов создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе основы разработки алгоритмов и программ для их практической реализации	Знает принципы структурного и объектно-ориентированного программирования, синтаксис и алфавит языков программирования Си++/Си, основные приемы алгоритмизации.
	ОПК-3.2. Умеет управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	Умеет использовать структурный и объектно-ориентированный подходы к разработке программ; умеет разрабатывать алгоритмы поставленных задач.
	ОПК-3.3. Владеет навыками разработки алгоритмов и программ, применяемых при создании и использовании продуктов и услуг в сфере информационно-коммуникационных технологий	Владеет навыками разработки алгоритмов и их реализации на языке Си/Си++ для решения задач профессиональной деятельности, в том числе с сфере ИКТ.
Профессиональные компетенции		
-	-	-

4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 академических часов.

Распределение трудоемкости дисциплины по видам учебной деятельности представлено в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины по видам учебной деятельности

Виды учебной деятельности	Всего часов	Семестры		
		1 семестр	2 семестр	3 семестр
Контактная аудиторная работа обучающихся с преподавателем, всего	48	14	14	20
Лабораторные занятия	28	8	8	12
Самостоятельная работа под руководством преподавателя	14	4	4	6
Контрольные работы	6	2	2	2
Самостоятельная работа обучающихся, в т.ч. контактная внеаудиторная работа обучающихся с преподавателем, всего	177	49	49	79
Самостоятельное изучение тем (вопросов) теоретической части дисциплины	97	21	25	51
Подготовка к контрольной работе	26	8	8	10

Согласована на портале № 65300

Подготовка к лабораторной работе	30	12	8	10
Написание отчета по лабораторной работе	24	8	8	8
Подготовка и сдача экзамена	27	9	9	9
Общая трудоемкость (в часах)	252	72	72	108
Общая трудоемкость (в з.е.)	7	2	2	3

5. Структура и содержание дисциплины

5.1. Разделы (темы) дисциплины и виды учебной деятельности

Структура дисциплины по разделам (темам) и видам учебной деятельности приведена в таблице 5.1.

Таблица 5.1 – Разделы (темы) дисциплины и виды учебной деятельности

Названия разделов (тем) дисциплины	Лаб. раб.	Контр. раб.	СРП, ч.	Сам. раб., ч	Всего часов (без промежуточной аттестации)	Формируемые компетенции
1 семестр						
1 Основы алгоритмизации	-	2	1	6	9	ОПК-3
2 Интегрированная среда программирования DEV-CPP	-		1	6	7	ОПК-3
3 Синтаксис и алфавит языка Си. Типы данных языка Си	-		1	8	9	ОПК-3
4 Подготовка и исполнение программы на языке Си. Конструкции структурного программирования в Си	8		1	29	38	ОПК-3
Итого за семестр	8	2	4	49	63	
2 семестр						
5 Функции	4	2	1	20	27	ОПК-3
6 Массивы	4		1	18	23	ОПК-3
7 Файлы в Си	-		1	6	7	ОПК-3
8 Управление выводом в консоль	-		1	5	6	ОПК-3
Итого за семестр	8	2	4	49	63	
3 семестр						
9 Введение в объектно-ориентированное программирование	-	2	1	14	17	ОПК-3
10 Введение в Си++	-		1	14	15	ОПК-3
11 Классы и объекты	8		2	22	32	ОПК-3
12 Наследование и полиморфизм	4		1	19	24	ОПК-3
13 Потоки ввода-вывода	-		1	10	11	ОПК-3
Итого за семестр	12	2	6	79	99	
Итого	28	6	14	177	225	

5.2. Содержание разделов (тем) дисциплины

Содержание разделов (тем) дисциплины приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов (тем) дисциплины

Названия разделов (тем) дисциплины	Содержание разделов (тем) дисциплины	СРП, ч	Формируемые компетенции
1 семестр			

Согласована на портале № 65300

1 Основы алгоритмизации	Основные понятия и определения. Типы данных. Структурное программирование. Системы кодирования алгоритмов. Система псевдокод. Блок-диаграммы. Диаграммы Насси − Шнейдермана. Основные алгоритмы. Алгоритмы суммы и произведения. Алгоритмы поиска	1	ОПК-3
	Итого	1	
2 Интегрированная среда программирования DEV-CPP	Подготовка программного обеспечения для работы в среде DEV-CPP. Настройка параметров среды. Создание проекта. Компиляция и выполнение. Отладка программы. Много-файловая компиляция. Сообщения об ошибках	1	ОПК-3
	Итого	1	
3 Синтаксис и алфавит языка Си. Типы данных языка Си	Алфавит языка Си. Синтаксис. Лексемы языка. Ключевые слова. Идентификаторы. Константы. Литеральные строки. Операторы. Знаки пунктуации. Основные типы данных. Простые типы. Приставки к типам данных. Преобразование типов. Производные типы данных. Указатели. Ссылки. Разыменование указателей. Сложные типы данных. Массивы. Структуры. Объединения. Перечисления. Объявления и инициализация переменных.	1	ОПК-3
	Итого	1	
4 Подготовка и исполнение программы на языке Си. Конструкции структурного программирования в Си	Этапы подготовки программы к исполнению. Директивы препроцессора. Директива #include. Директива #include_next. Директивы #define, #undef, #ifdef, #ifndef. Директивы условной компиляции. Управляющая директива #line. Директива #error. Директива #pragma. Ввод-вывод информации. Функция printf. Функция scanf. Простая программа на языке Си. Следование. Ветвление. Оператор проверки условия if <else>. Множественный выбор. Циклы. Цикл с фиксированным числом операций for. Циклы while и do while. Операторы безусловной передачи управления continue и break. Примеры использования операторов цикла. Вычисление суммы бесконечного ряда. Вычисления по итерационной формуле. Программирование численных методов	1	ОПК-3
	Итого	1	
	Итого за семестр	4	
2 семестр			

Согласована на портале № 65300

5 Функции	Синтаксис. Объявление и вызов функций. Локальные переменные. Выход из функций. Передача параметров по ссылке. Рекурсивные функции	1	ОПК-3
	Итого	1	
6 Массивы	Одномерные массивы. Инициализация массива. Поиск значений в массиве. Сортировка массивов. Многомерные массивы. Инициализация матриц. Печать матриц. Примеры решений задач с использованием матриц. Строки. Инициализация строк. Представление строки в памяти компьютера. Стандартные функции для работы со строками	1	ОПК-3
	Итого	1	
7 Файлы в Си	Типы файлов в Си. Механизм чтения-записи. Функции для поточного доступа к файлам. Примеры работы с текстовыми файлами. Запись данных в текстовый файл. Чтение данных из текстового файла. Изменение текстового файла. Двоичные файлы. Запись и чтение информации в двоичный файл. Реализация прямого доступа в двоичном файле	1	ОПК-3
	Итого	1	
8 Управление выводом в консоль	Win32 API. Типы данных Windows. Функции WinAPI	1	ОПК-3
	Итого	1	
Итого за семестр		4	
3 семестр			
9 Введение в объектно-ориентированное программирование	Основные положения. Основные принципы объектно-ориентированного программирования. Отношения между объектами. Причины развития объектно-ориентированного подхода к программированию	1	ОПК-3
	Итого	1	
10 Введение в Си++	Связь Си и Си++. Классы. Элементы класса. Атрибуты доступа. Реализация методов класса. Ввод-вывод в Си++. Простая программа на языке Си++.	1	ОПК-3
	Итого	1	
11 Классы и объекты	Конструкторы. Деструкторы. Указатель this. Статические элементы класса. Перегрузка операций. Дружественные функции	2	ОПК-3
	Итого	2	
12 Наследование и полиморфизм	Базовые и производные классы. Полиморфизм. Абстрактные классы. Типы наследования	1	ОПК-3
	Итого	1	

Согласована на портале № 65300

13 Потоки ввода-вывода	Иерархия потоковых классов Си++. Консольный ввод-вывод. Форматирование вывода. Файловый ввод вывод	1	ОПК-3
	Итого	1	
Итого за семестр		6	
Итого		14	

5.3. Контрольные работы

Виды контрольных работ и часы на контрольные работы приведены в таблице 5.3.

Таблица 5.3 – Контрольные работы

№ п.п.	Виды контрольных работ	Трудоемкость, ч	Формируемые компетенции
1 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ОПК-3
Итого за семестр		2	
2 семестр			
2	Контрольная работа с автоматизированной проверкой	2	ОПК-3
Итого за семестр		2	
3 семестр			
3	Контрольная работа с автоматизированной проверкой	2	ОПК-3
Итого за семестр		2	
Итого		6	

5.4. Лабораторные занятия

Наименование лабораторных работ приведено в таблице 5.4.

Таблица 5.4 – Наименование лабораторных работ

Названия разделов (тем) дисциплины	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
1 семестр			
4 Подготовка и исполнение программы на языке Си. Конструкции структурного программирования в Си	Проверка условий	4	ОПК-3
	Обработка массивов	4	ОПК-3
	Итого	8	
Итого за семестр		8	
2 семестр			
5 Функции	Обработка матриц. Функции	4	ОПК-3
	Итого	4	
6 Массивы	Работа со строками	4	ОПК-3
	Итого	4	
Итого за семестр		8	
3 семестр			

Согласована на портале № 65300

11 Классы и объекты	Реализация класса.	4	ОПК-3
	Конструкторы и деструкторы	4	ОПК-3
	Итого	8	
12 Наследование и полиморфизм	Перегрузка операторов.	4	ОПК-3
	Итого	4	
Итого за семестр		12	
Итого		28	

5.5.Контроль самостоятельной работы (курсовой проект / курсовая работа)

Не предусмотрено учебным планом

5.6. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 5.6.

Таблица 5.6 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов (тем) дисциплины	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
1 семестр				
1 Основы алгоритмизации	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	4	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	6		
2 Интегрированная среда программирования DEV-CPP	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	4	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	6		
3 Синтаксис и алфавит языка Си. Типы данных языка Си	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	6	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	8		

Согласована на портале № 65300

4 Подготовка и исполнение программы на языке Си. Конструкции структурного программирования в Си	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	7	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Подготовка к лабораторной работе	12	ОПК-3	Лабораторная работа
	Написание отчета по лабораторной работе	8	ОПК-3	Отчет по лабораторной работе
	Итого	29		
Итого за семестр		49		
	Подготовка и сдача экзамена	9		Экзамен
2 семестр				
5 Функции	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	10	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Подготовка к лабораторной работе	4	ОПК-3	Лабораторная работа
	Написание отчета по лабораторной работе	4	ОПК-3	Отчет по лабораторной работе
	Итого	20		
6 Массивы	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	8	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Подготовка к лабораторной работе	4	ОПК-3	Лабораторная работа
	Написание отчета по лабораторной работе	4	ОПК-3	Отчет по лабораторной работе
	Итого	18		

Согласована на портале № 65300

7 Файлы в Си	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	4	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	6		
8 Управление выводом в консоль	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	3	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	5		
Итого за семестр		49		
	Подготовка и сдача экзамена	9		Экзамен
3 семестр				
9 Введение в объектно-ориентированное программирование	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	12	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	14		
10 Введение в Си++	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	12	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	14		

Согласована на портале № 65300

11 Классы и объекты	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	12	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Подготовка к лабораторной работе	4	ОПК-3	Лабораторная работа
	Написание отчета по лабораторной работе	4	ОПК-3	Отчет по лабораторной работе
	Итого	22		
12 Наследование и полиморфизм	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	7	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Подготовка к лабораторной работе	6	ОПК-3	Лабораторная работа
	Написание отчета по лабораторной работе	4	ОПК-3	Отчет по лабораторной работе
	Итого	19		
13 Потоки ввода-вывода	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	8	ОПК-3	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-3	Контрольная работа
	Итого	10		
Итого за семестр		79		
	Подготовка и сдача экзамена	9		Экзамен
Итого		204		

5.7. Соответствие компетенций, формируемых при изучении дисциплины, и видов учебной деятельности

Соответствие компетенций, формируемых при изучении дисциплины, и видов учебной деятельности представлено в таблице 5.7.

Таблица 5.7 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Формируемые компетенции	Виды учебной деятельности				Формы контроля
	Лаб. раб.	Конт.Раб.	СРП	Сам. раб.	

Согласована на портале № 65300

ОПК-3	+	+	+	+	Контрольная работа, Лабораторная работа, Отчет по лабораторной работе, Тестирование, Экзамен

6. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература

1. Пермякова Н.В. Информатика и программирование [Электронный ресурс]: учебное пособие.Томск : ФДО, ТУСУР, 2016. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).
2. Пермякова Н.В. Программирование на языке высокого уровня [Электронный ресурс]: Учебное пособие. В 2-х частях. — Томск: Томский межвузовский центр дистанционного образования, 2007. — Ч.2. — 126 с. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).

7.2. Дополнительная литература

1. М.Ю. Катаев. Спецкурс−1. Высокоуровневые методы информатики и программирования [Электронный ресурс]: Учебное пособие. – Томск: Томский межвузовский центр дистанционного образования, 2006. − 132 с. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).
2. Борисов С.И. Объектно-ориентированное программирование [Электронный ресурс]: Учебное пособие. − Томск: Томский межвузовский центр дистанционного образования, 2002. − 217 с. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).

7.3. Учебно-методические пособия

7.3.1. Обязательные учебно-методические пособия

1. Пермякова Н.В. Информатика и программирование [Электронный ресурс]: лабораторный практикум. -Томск : ФДО ТУСУР, 2016. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).
2. Пермякова Н.В. Информатика и программирование [Электронный ресурс]: методические указания по организации самостоятельной работы для студентов заочной формы обучения направления подготовки 38.03.05 Бизнес-информатика, обучающихся с применением дистанционных образовательных технологий / Н. В. Пермякова. Ю. В. Морозова . – Томск : ФДО, ТУСУР, 2018. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).
3. Пермякова Н. В. Информатика и программирование [Электронный ресурс]: методические указания по выполнению лабораторных работ. – Томск : ФДО ТУСУР, 2016. Режим доступа: https://study.tusur.ru/study/library/ (доступ из личного кабинета студента).

7.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:
– в форме электронного документа;
– в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:
– в форме электронного документа;
– в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:
– в форме электронного документа;
– в печатной форме.

Согласована на портале № 65300

7.4. Иное учебно-методическое обеспечение

1. Пермякова Н. В. Информатика и программирование [Электронный ресурс]: электронный курс / Н.В. Пермякова. – Томск : ФДО ТУСУР, 2016. (доступ из личного кабинета студента).
2. Пермякова Н. В. Программирование С++ [Электронный ресурс]: электронный курс / Н.В. Пермякова. – Томск : ФДО ТУСУР, 2014. (доступ из личного кабинета студента).

7.5. Современные профессиональные базы данных и информационные справочные системы

1. При изучении дисциплины рекомендуется обращаться к современным базам данных, информационно-справочным и поисковым системам, к которым у ТУСУРа открыт доступ: https://lib.tusur.ru/ru/resursy/bazy-dannyh.

8. Материально-техническое и программное обеспечение дисциплины

8.1. Общие требования к материально-техническому и программному обеспечению дисциплины

Учебные аудитории для проведения занятий лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, для самостоятельной работы студентов

634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- Веб-камера - 6 шт.;
- Наушники с микрофоном - 6 шт.;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- 7-Zip;
- DEV C++ (с возможностью удаленного доступа);
- Google Chrome;
- Kaspersky Endpoint Security для Windows;
- LibreOffice 7.0.6.2;
- Microsoft Windows;
- Notepad++ (с возможностью удаленного доступа);

8.2. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 209 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- учебная мебель;
- компьютеры;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду ТУСУРа.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

8.3. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Согласована на портале № 65300

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися с **нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися с **нарушениями зрения** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися с **нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

9. Оценочные материалы и методические рекомендации по организации изучения дисциплины

9.1. Содержание оценочных материалов для текущего контроля и промежуточной аттестации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы, представленные в таблице 9.1.

Таблица 9.1 – Формы контроля и оценочные материалы

Названия разделов (тем) дисциплины	Формируемые компетенции	Формы контроля	Оценочные материалы (ОМ)
1 Основы алгоритмизации	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
2 Интегрированная среда программирования DEV-CPP	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
3 Синтаксис и алфавит языка Си. Типы данных языка Си	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов

Согласована на портале № 65300

4 Подготовка и исполнение программы на языке Си. Конструкции структурного программирования в Си	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Лабораторная работа	Темы лабораторных работ
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
		Отчет по лабораторной работе	Темы лабораторных работ
5 Функции	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Лабораторная работа	Темы лабораторных работ
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
		Отчет по лабораторной работе	Темы лабораторных работ
6 Массивы	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Лабораторная работа	Темы лабораторных работ
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
		Отчет по лабораторной работе	Темы лабораторных работ
7 Файлы в Си	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов

Согласована на портале № 65300

8 Управление выводом в консоль	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
9 Введение в объектно-ориентированное программирование	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
10 Введение в Си++	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
11 Классы и объекты	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Лабораторная работа	Темы лабораторных работ
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
		Отчет по лабораторной работе	Темы лабораторных работ
12 Наследование и полиморфизм	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Лабораторная работа	Темы лабораторных работ
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
		Отчет по лабораторной работе	Темы лабораторных работ

Согласована на портале № 65300

13 Потоки ввода-вывода	ОПК-3	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов

Шкала оценки сформированности отдельных планируемых результатов обучения по дисциплине приведена в таблице 9.2.

Таблица 9.2 – Шкала оценки сформированности планируемых результатов обучения по дисциплине

Оценка	Баллы за ОМ	Формулировка требований к степени сформированности планируемых результатов обучения		
		знать	уметь	владеть
2 (неудовлетворительно)	< 60% от максимальной суммы баллов	отсутствие знаний или фрагментарные знания	отсутствие умений или частично освоенное умение	отсутствие навыков или фрагментарные применение навыков
3 (удовлетворительно)	от 60% до 69% от максимальной суммы баллов	общие, но не структурированные знания	в целом успешно, но не систематически осуществляемое умение	в целом успешное, но не систематическое применение навыков
4 (хорошо)	от 70% до 89% от максимальной суммы баллов	сформированные, но содержащие отдельные проблемы знания	в целом успешное, но содержащие отдельные пробелы умение	в целом успешное, но содержащие отдельные пробелы применение навыков
5 (отлично)	≥ 90% от максимальной суммы баллов	сформированные систематические знания	сформированное умение	успешное и систематическое применение навыков

Шкала комплексной оценки сформированности компетенций приведена в таблице 9.3.

Таблица 9.3 – Шкала комплексной оценки сформированности компетенций

Оценка	Формулировка требований к степени компетенции
2 (неудовлетворительно)	Не имеет необходимых представлений о проверяемом материале или Знать на уровне **ориентирования**, представлений. Обучающийся знает основные признаки или термины изучаемого элемента содержания, их отнесенность к определенной науке, отрасли или объектам, узнает в текстах, изображениях или схемах и знает, к каким источникам нужно обращаться для более детального его усвоения.
3 (удовлетворительно)	Знать и уметь на **репродуктивном** уровне. Обучающихся знает изученный элемент содержания репродуктивно: произвольно воспроизводит свои знания устно, письменно или в демонстрируемых действиях.

Согласована на портале № 65300

4 (хорошо)	Знать, уметь, владеть на **аналитическом** уровне. Зная на репродуктивном уровне, указывать на особенности и взаимосвязи изученных объектов, на их достоинства, ограничения, историю и перспективы развития и особенности для разных объектов усвоения.
5 (отлично)	Знать, уметь, владеть на **системном** уровне. Обучающийся знает изученный элемент содержания системно, произвольно и доказательно воспроизводит свои знания устно, письменно или в демонстрируемых действиях, учитывая и указывая связи и зависимости между этим элементом и другими элементами содержания дисциплины, его значимость в содержании дисциплины.

9.1.1. Примерный перечень тестовых заданий

1. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования. Какой тип передачи управления может использовать такая программа?
   безусловный
   объектно-зависимый
   условный
   функционально-зависимый

2. Управление информацией может быть организовано с использованием различных программных
   средств, одним из которых являются языки программирования. Какая из перечисленных констант записана верно с точки зрения синтаксиса языка программирования Си?
   5,025
   -12e-0.12
   0197
   5.

3. Для управления информацией, хранящейся в цифровом виде можно использовать самостоятельно
   разработанные программные средства. Процесс разработки таких средств подразумевает владение языком программирования. Какое имя идентификатора переменной, записано неверно,
   с точки зрения синтаксиса языка программирования Си?
   2a
   func
   a_b
   A2

4. Информация, хранящаяся на компьютере, может обрабатываться в оперативной памяти. Язык программирования Си допускает в явном виде работу с адресами физической памяти – указателями.
   Какая из перечисленных переменных описана как указатель?
   int p[25];
   int * f;
   int z[12][3];
   int &a;

5. Информация, хранящаяся на компьютере, может обрабатываться в оперативной памяти. Язык программирования Си допускает в явном виде работу с адресами физической памяти.
   После выполнения какого из перечисленных фрагментов кода в переменной x будет храниться
   адрес переменной y?

Согласована на портале № 65300

```
int *x;
int y = 15;
x = *y;
------------
int *x;
int y = 15;
x = y*;
------------
int *x;
int y = 15;
x = &y;
------------
int *x;
int y = 15;
x = #y;
------------
```

6. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования, фрагмент кода которой приведен ниже.

```
int x = 9;
int y = 4;
int z = 2;
if (x==y && z>0) {z = x;
x = y;
y = z;}
else { z = x*2;
x = 2*y;
y = z;}
printf (“ %d %d %d”, x,y,z);
```

Что будет выведено на экран выполнении этого фрагмента?

18 8 18
9 4 9
4 9 9
8 18 18

7. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования, фрагмент кода которой приведен ниже. Что будет выведено
на экран при выполнении этого фрагмента?

```
int i = 25;
do{
printf("%3d",i);
i-=2;
}
while(i>=13);
```

25 23 21 19 17 15 13
25 23 21 19 17 15
13 15 17 19 21 23 25
13 15 17 19 21 23

8. Перед программистом была поставлена задача получения числовой информации в виде последовательности:
2 5 8 11 14 17 20.
Какой из представленных фрагментов решает поставленную задачу?

```
int i = 2;
do{
printf("%3d",i);
```

Согласована на портале № 65300

```
i+=3;
}
while(i<20);
----------------
int i = 2;
while(i<=20){
printf("%3d",i);
i+=3;
}
--------------------
int i = 2;
while(i>=20){
printf("%3d",i);
i+=3;
}
------------------
int i = 2;
do{
printf("%3d",i);
i+=3;
}
while(i<25);
------------------
```

9. Перед программистом поставлена задача получения числовой информации в виде последовательности:
   10.0000 5.0000 2.5000 1.2500 0.625 0.3125?
   Какой из представленных циклов может быть использован для решения этой задачи?

```
---------------
float i;
for (i=10; i>0.2; i-=5)
printf("%.4f ", i);
---------------
float i;
for (i=10; i>0.2; i--)
printf("%.4f ", i);
-----------------
float i;
for (i=10; i>0.2; i/=2)
printf("%.4f ", i);
-----------------
float i;
for (i=0.3125; i<11; i*=2)
printf("%.4f ", i);
---------------------
```

10. Для решения некоторой задачи обработки массива была использована программа, представленная
    ниже:

```
int main(int argc, char *argv[])
{
system("chcp 1251");
int x[10] = {2,7,6,1,9,5,8,3,4,0};
int k = 0,i;
for (i=0;i<10;i++)
if (x[i]%2==0) printf("%3d",i);
printf("\n");
```

Согласована на портале № 65300

```
system("PAUSE");
return EXIT_SUCCESS;
}
```

Какую задачу решает написанная программа?
Печать индексов нечетных по значению элементов массива
Поиск суммы четных по значению элементов массива
Поиск суммы нечетных по значению элементов массива
Печать индексов четных по значению элементов массива

11. Компьютерная программа, код которой представлен ниже, была использована для обработки
числового массива. Что будет выведено на экран при выполнении программы?

```
int main(int argc, char *argv[])
{
system("chcp 1251");
int x[10] = {6, 9, 2, 3, 7,
7, 6, 11, 8, 5};
int k = x[0],i;
for (i=1;i<10;i++)
if (x[i]>k) k = x[i];
printf("%3d",k);
printf("\n");
system("PAUSE");
return EXIT_SUCCESS;
}
```

11
2
7
5

12. Компьютерная программа, код которой представлен ниже, была использована для обработки
числового массива. В каком порядке будут расположены элементы массива после выполнения
программы?

```
int main(int argc, char *argv[])
{
system("chcp 1251");
int x[10] = {6, 9, 2, 3, 7,
7, 6, 11, 8, 5};
int i,j,k;
int m = 3;
for (j=0;j<m;j++){
k = x[9];
for (i=9;i>0;i--)
x[i] = x[i-1];
x[0] = k;
}
for (i=0;i<10;i++)
printf("%3d",x[i]);
printf("\n");
system("PAUSE");
return EXIT_SUCCESS;}
```

6 9 2 3 7 7 6 11 8 5
11 8 5 6 9 2 3 7 7 6
8 5 6 9 2 3 7 7 6 11
5 6 9 2 3 7 7 6 11 8

Согласована на портале № 65300

13. Компьютерная программа, код которой представлен ниже, была использована для обработки
числового двумерного массива. Что будет выведено на экран при выполнении программы?

```
int main(int argc, char *argv[])
{
int A[5][5]={1,2,3,4,5,
6,7,8,9,10,
11,12,13,14,15,
16,17,18,19,20,
21,22,23,24,25};
int i,j,k,n=5;
k = A[1][0];
for(i=0;i<n;i++)
for(j=0;j<i;j++)
if (A[i][j]>k) k=A[i][j];
printf("%d\n",k);
system("PAUSE");
return 0;
}
```

6
10
24
25

14. Обрабатываемая в задаче информация была представлена в виде класса языка Си. Что произойдет при компиляции и выполнении?

```
class A {
int b;
public:
A(int i) {b = i;}
A() {b = 111;}
void Show(){ cout<<b;}
}
void main(){
A* obj = new A();
obj ->show();
}
```

напечатается 1
напечатается 111
ошибка компиляции
ошибка выполнения

15. При обработке информации данные были описаны с помощью классов языка Си. Каким атрибутом доступа должен быть описан элемент класса, который можно использовать только внутри
класса?
public
static
private
protected

16. При обработке информации данные были описаны с помощью классов языка Си. Какое из следующих утверждений является верным?
В классе может быть только один конструктор
В классе может быть только один деструктор

Согласована на портале № 65300

В классе обязательно должен быть написан хотя бы один конструктор
Конструктор класса вызывается при каждом обращении к объекту

17. Обрабатываемая в задаче информация была представлена в виде класса языка Си. Что произойдет при компиляции и выполнении?
```
class A {
int b;
public:
A(int i) {b = i;}
A() {b = 0;}
void Show(){ cout<<b;}
}
void main(){
A* obj = new A(1);
cout <<obj->b;
}
```
напечатается 0
напечатается 1
ошибка компиляции
ошибка выполнения

18. При обработке информации данные были описаны с помощью классов языка Си. Каким атрибутом доступа должен быть описан элемент класса, который можно использовать извне класса?
public
static
private
protected

19. При обработке информации данные были описаны с помощью классов языка Си. Какое из следующих утверждений является верным?
Если в классе нет описания конструктора, то создается конструктор по умолчанию
Статические поля могут принимать значения, разные для каждого из объектов класса
Конструктор класса обязательно должен быть описан атрибутом доступа public
В классе может быть описано несколько конструкторов с одинаковыми списками параметров

20. При обработке информации данные были описаны с помощью класса языка Си. class
```
Point{
private:
int x;
int y;
public:
Point (int x1, int y1);
Point();
void Show();
int GetX();
int GetY();
void PutX(int X);
void PutY(int Y);
};
```
Какой из перечисленных ниже элементов класса является конструктором класса:
```
int x;
Point(int x1, int x2);
void Show();
int GetX();
```

Согласована на портале № 65300

9.1.2. Перечень экзаменационных вопросов

Приведены примеры типовых заданий из банка экзаменационных тестов, составленных по пройденным
разделам дисциплины.

1. Какую задачу решает представленный алгоритм?
   алг Поиск нач
   дано цел X[10]
   цел n=10
   цел a:=X[1]
   для i от 2 до n нц
   если X[i] < a то a:=X[i]
   кц
   рез a
   кон
   1. Поиск минимального элемента массива.
   2. Поиск индекса первого минимального элемента массива.
   3. Поиск индекса последнего минимального элемента массива.
   4. Поиск максимального элемента массива.

2. Какую задачу решает представленный алгоритм?
   алг Поиск нач
   дано цел X[10]
   цел n=10
   цел b:=1
   для i от 2 до n нц
   если X[i]<X[b] то b:=I
   кц
   рез b
   кон
   1. Поиск минимального элемента массива.
   2. Поиск индекса первого минимального элемента массива.
   3. Поиск индекса последнего минимального элемента массива.
   4. Поиск максимального элемента массива.

3. Какую задачу решает представленный алгоритм?
   алг Поиск нач
   дано цел X[10]
   цел n=10
   цел b:=1
   для i от 2 до n нц
   если X[i] ≤ X[b] то b:=I
   кц
   рез b
   кон
   1. Поиск минимального элемента массива.
   2. Поиск индекса первого минимального элемента массива.
   3. Поиск индекса последнего минимального элемента массива.
   4. Поиск максимального элемента массива.

4. Какую задачу решает представленный алгоритм?
   алг Вопрос4 нач
   дано цел X[10]
   цел n=10
   цел a:=0
   для i от 1 до n нц
   если X[i] > 0 то a:=a+X[i]

Согласована на портале № 65300

кц
рез a
кон
1. Поиск суммы положительных элементов.
2. Поиск суммы индексов положительных элементов.
3. Поиск количества положительных элементов.
4. Поиск суммы отрицательных элементов.

5. Какую задачу решает представленный алгоритм?
алг Вопрос5 нач
дано цел X[10]
цел n=10
цел a:=0
для i от 1 до n нц
если X[i] > 0 то a:=a+i
кц
рез a
кон
1. Поиск суммы положительных элементов.
2. Поиск суммы индексов положительных элементов.
3. Поиск количества положительных элементов.
4. Поиск суммы отрицательных элементов.

6. Программисту требовалось написать алгоритм поиска индекса первого элемента массива, имеющего положительное значение. Программист записал алгоритм следующим образом.
1. цел таб A[n]
2. для i от 1 до n нц
3. ввод A[i]
4. кц
5. для i от 1 до n нц
6. если A[i]>0 то p:=I
7. i:=n
8. кц
9. рез p
1. Алгоритм написан верно.
2. В строке 5.
3. В строке 6.
4. В строке 7.

7. Программисту требовалось написать алгоритм поиска индекса первого элемента массива, имеющего положительное значение. Программист записал алгоритм следующим образом.
1. цел таб A[n]
2. для i от 1 до n нц
3. ввод A[i]
4. кц
5. для i от 1 до n нц
6. если A[i]>0 то p:=I
7. i:=i+1
8. кц
9. рез p
1. Алгоритм написан верно.
2. В строке 5.
3. В строке 6.
4. В строке 7..

8. Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива

X ={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} ?
n=10
цел таб X[n]
для i от 1 до n нц
ввод X[i]
кц
p:=1
для i от 2 до n нц
если X[p] > X[i] то p :=i
кц
рез p
1. Алгоритм находит индекс первого минимального элемента.
2. Алгоритм находит индекс последнего минимального элемента.
3. Алгоритм находит индекс первого максимального элемента.
4. Алгоритм находит индекс последнего максимального элемента.

9. Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива
X ={-1, -2, -3, -4, -5, -6, -7, -8, -9, -10} ?
n=10
цел таб X[n]
для i от 1 до n нц
ввод X[i]
кц
p:=1
для i от 2 до n нц
если X[p] >= X[i] то p :=i
кц
рез p
1. Алгоритм находит индекс первого минимального элемента.
2. Алгоритм находит индекс последнего минимального элемента.
3. Алгоритм находит индекс первого максимального элемента.
4. Алгоритм находит индекс последнего максимального элемента.

10. Укажите описание соответствующее фрагменту программы:
```
int X[10];
int i;
for (i=0;i<10;i++)
X[i]=i+2;
```
1. Описание и инициализация статического целочисленного массива из 10-ти элементов.
2. Описание и инициализация динамического вещественного массива из 10 –ти элементов.
3. Описание и инициализация статического вещественного массива из 10-ти элементов.
4. Описание и инициализация динамического целочисленного массива из 10 –ти элементов.

11. Укажите описание соответствующее фрагменту программы:
```
int *X = (int*)malloc(sizeof(int)*10);
int I;
for (i=0;i<10;i++)
X[i]=i+2;
```
1. Описание и инициализация статического целочисленного массива из 10-ти элементов.
2. Описание и инициализация динамического вещественного массива из 10 –ти элементов.
3. Описание и инициализация статического вещественного массива из 10-ти элементов.
4. Описание и инициализация динамического целочисленного массива из 10 –ти элементов.

Согласована на портале № 65300

12. Выберите верные утверждения:
    1. Оператор «->» - непрямой селектор компоненты.
    2. Оператор «++» - бинарный оператор.
    3. Оператор «%» - оператор целочисленного деления.
    4. Оператор «=» - оператор «равно».

13. Выберите верные утверждения:
    1. Оператор «++» - оператор префиксного (или постфиксного увеличения).
    2. Оператор «^» - вычисление степени.
    3. Оператор «%» - оператор целочисленного деления.
    4. Оператор «*» - оператор прямой адресации.

14. Выберите верные утверждения:
    1. В последовательности int m; float b[12]; char* x; 13 лексем.
    2. В последовательности int m; float b[12]; char* x; 11 лексем.
    3. Последовательности float k,i; char z[12]; и float k, I; char z[12]; лексически не эквивалентны.
    4. В последовательности int a,b; float * x; 7 лексем.

15. Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива
    X ={10, 9, 8, 7, 6, 5, 6, 7, 8, 2, 1} ?
    n=10
    цел таб X[n]
    для i от 1 до n нц
    ввод X[i]
    кц
    p:=1
    для i от 2 до n нц
    если X[p] < X[i] то p :=i
    кц
    рез p
    1. Алгоритм находит индекс первого минимального элемента.
    2. Алгоритм находит индекс последнего минимального элемента.
    3. Алгоритм находит индекс первого максимального элемента.
    4. Алгоритм находит индекс последнего максимального элемента.

16. Выберите верные утверждения:
    1. Алгоритм должен удовлетворять требованию конечности записи.
    2. Алгоритм должен удовлетворять требованию последовательности записи.
    3. Алгоритм должен удовлетворять требованию дискретности действий.
    4. Алгоритм должен удовлетворять требованию структурности.

17. Выберите верные утверждения:
    1. Алгоритм должен содержать конечное количество шагов.
    2. Алгоритм должен содержать заданное количество шагов.
    3. Алгоритм должен выполнять заданное количество шагов при решении задачи.
    4. Алгоритм должен быть дискретным для всех допустимых исходных данных.

18. Пяти программистам дали задания написать алгоритм суммирования целых чисел A1, A2, …, An. Некоторые программисты допустили ошибки в алгоритме. Какой фрагмент записан верно?
    1.
    цел S:=0
    для I от A1 до An нц
    S:=S+i

Согласована на портале № 65300

кц
рез S
2.
цел S:=A1
для I от A1 до An нц
S:=S+i
кц
рез S
3.
цел S:=0
i:=A1
пока I < An нц
S:=S+i
i:=i+1;
кц
рез S
4.
цел S:=0
i:=An
пока I ≥ A1 нц
S:=S+i
i:=i+1;
кц
рез S

19. Пяти программистам дали задания написать алгоритм произведения целых чисел A1, A2, …, An. Некоторые программисты допустили ошибки в алгоритме. Какой фрагмент записан верно?
1.
цел S:=1
для I от A1 до An нц
S:=S*i
кц
рез S
2.
цел S:=0
для I от A1 до An нц
S:=S*i
кц
рез S
3.
цел S:=1
i:=A1
пока I < An нц
S:=S*i
i:=i+1;
кц
рез S
4.
цел S:=0
i:=An
пока I ≥ A1 нц
S:=S*i
i:=i+1;
кц
рез S

Согласована на портале № 65300

20. Выберите верно записанные идентификаторы:
    1. sinus
    2. 2Summa
    3. Sinu*s
    4. double

9.1.3. Примерный перечень тем и тестовых заданий на контрольные работы

Информатика и программирование

1. Что будет выведено на экран при выполнении следующего фрагмента программы
   int i;
   for(i=1;i<10;i+=2) printf("%d ",i);
   1. Вывод на экран чисел 1, 3, 5, 7, 9
   2. Вывод на экран чисел 1, 4, 9, 16, 25
   3. Вывести на экран числа 3,5,7,9,11
   4. Вывод на экран чисел 3,5,7,9,11

2. Что будет выведено на экран при выполнении следующего фрагмента программы
   int i;
   for(i=0;i<5;i++) printf("%d ",(i+1)*(i+1));
   1. Вывод на экран чисел 1, 3, 5, 7, 9
   2. Вывод на экран чисел 1, 4, 9, 16, 25
   3. Вывести на экран числа 3,5,7,9,11
   4. Вывод на экран чисел 3,5,7,9,11

3. Выберите фрагменты программ, в которых цикл выполняется 7 раз.
   1. int i=1; while (i<20) i+=3;
   2. int i=0; while (i<15)i+=3;
   3. int i=0; while (i<12)i+=2;
   4. int i=4; while (i<40) i+=i;

4. Что будет выведено на экран при выполнении следующего фрагмента программы при n = 2500?
   ```
   int func1(int p){
   int s=0;
   while(p>0){s+=p%10;
   p/=10;
   }
   return s; }
   int main(int argc, char *argv[])
   { int n;
   scanf("%d",&n);
   printf("%d\n",func1(n));
   system("PAUSE");
   return 0;}
   ```
   1. На экран выведется сумма цифр числа.
   2. На экран выведется максимальная цифра числа.
   3. На экран выведется минимальная цифра числа.
   4. На экран выведется количество цифр числа.

5. Что будет выведено на экран при выполнении следующего фрагмента программы при n = 1000 , m = 5?
   ```
   int func13(int p, int v){
   int s=0,t;
   while(p>0){t=p%10;
   if(t==v)s++;
   p/=10;
   ```

```
}
return s; }
int main(int argc, char *argv[])
{ int n,m;
scanf("%d%d",&n,&m);
printf("%d\n",func13(n,m));
system("PAUSE");
return 0;}
```

1. На экран выведется количество цифр числа n равных m.
2. На экран выведется количество цифр числа n не равных m.
3. На экран выведется количество цифр числа n больших m.
4. На экран выведется количество цифр числа n меньших m.

6. Что будет выведено на экран при выполнении следующего фрагмента программы при n = 2020 , m =7?

```
int func16(int p, int v){
int s=0,t;
while(p>0){t=p%10;
if(t<v)s++;
p/=10;
}
return s; }
int main(int argc, char *argv[])
{ int n,m;
scanf("%d%d",&n,&m);
printf("%d\n",func16(n,m));
system("PAUSE");
return 0;}
```

1. На экран выведется количество цифр числа n равных m.
2. На экран выведется количество цифр числа n не равных m.
3. На экран выведется количество цифр числа n больших m.
4. На экран выведется количество цифр числа n меньших m.

7. Чему будет равно значение переменной p при выполнении следующей программы, если элементы
массива y заданы следующим образом y = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}?

```
int Function1(int *x, int f, int l, int z) {
int S = 0,i;
for(i=f;i<l;i+=z) S+=x[i];
return S; }
int main(int argc, char *argv[])
{ int y[10],i,p;
for(i=0;i<10;i++)
y[i]=rand()%11-rand()%11;
p = Function1(y,0,10,1);
return 0; }
```

1. p - сумма всех элементов массива y.
2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

8. Чему будет равно значение переменной p при выполнении следующей программы, если элементы
массива y заданы следующим образом y = {-10, 2, 3, 4, 5, 6, 7, 8, 9, 1}?

```
p = Function1(y,1,20,2);
int Function1(int *x, int f, int l, int z) {
int S = 0,i;
```

Согласована на портале № 65300

```
for(i=f;i<l;i+=z) S+=x[i];
return S; }
int main(int argc, char *argv[])
{ int y[10],i,p;
for(i=0;i<10;i++)
y[i]=rand()%11-rand()%11;
p = Function1(y,1,10,2);
return 0;}
```
1. p - сумма всех элементов массива y.
2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

9. Чему будет равно значение переменной p при выполнении следующей программы, если элементы
массива y заданы следующим образом y = {-10, 2, 3, 4, 5, 6, 7, 8, 9, 1}?
```
p = Function1(y,1,20,2);
int Function1(int *x, int f, int l, int z) {
int S = 0,i;
for(i=f;i<l;i+=z) S+=x[i];
return S; }
int main(int argc, char *argv[])
{ int y[10],i,p;
for(i=0;i<10;i++)
y[i]=rand()%11-rand()%11;
p = Function1(y,1,10,2);
return 0;}
```
1. p - сумма всех элементов массива y.
2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

10. Сколько шагов выполнит внешний шаг сортировки обменом при сортировке элементов массива
X = {-5, 3, 0, -7, 10, -3, -5, 1, 2, 5}?
```
int main() {
int n = 10, X[n],i,C=0,p=0,v,j;
srand(time(NULL));
for(i=0;i<10;i++){
X[i]=rand()%11-rand()%11;
printf("%d ",X[i]);}
for(i=0;i<n-1;i++){
p=0;C++;
for(j=0;j<n-i-1;j++)
{ if(X[j]>X[j+1])
{ v=X[j];
X[j]=X[j+1];
X[j+1]=v;
p++;}}
if (p==0)break; }
printf("\n");
for(i=0;i<10;i++)
printf("%d ",X[i]);
printf("\n %d \n",C);
system("PAUSE");
return 0;
}
```

Согласована на портале № 65300

1. 10
2. 8
3. 5
4. 6

9.1.4. Темы лабораторных работ

1. Проверка условий
2. Обработка массивов
3. Обработка матриц. Функции
4. Работа со строками
5. Реализация класса.
6. Конструкторы и деструкторы
7. Перегрузка операторов.

9.2. Методические рекомендации

Учебный материал излагается в форме, предполагающей самостоятельное мышление студентов, самообразование. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Начать изучение дисциплины необходимо со знакомства с рабочей программой, списком учебно-методического и программного обеспечения. Самостоятельная работа студента включает работу с учебными материалами, выполнение контрольных мероприятий, предусмотренных учебным планом.

В процессе изучения дисциплины для лучшего освоения материала необходимо регулярно обращаться к рекомендуемой литературе и источникам, указанным в учебных материалах; пользоваться через кабинет студента на сайте Университета образовательными ресурсами электронно-библиотечной системы, а также общедоступными интернет-порталами, содержащими научно-популярные и специализированные материалы, посвященные различным аспектам учебной дисциплины.

При самостоятельном изучении тем следуйте рекомендациям:

– чтение или просмотр материала осуществляйте со скоростью, достаточной для индивидуального понимания и освоения материала, выделяя основные идеи; на основании изученного составить тезисы. Освоив материал, попытаться соотнести теорию с примерами из практики;

– если в тексте встречаются незнакомые или малознакомые термины, следует выяснить их значение для понимания дальнейшего материала;

– осмысливайте прочитанное и изученное, отвечайте на предложенные вопросы.

Студенты могут получать индивидуальные консультации, в т.ч. с использованием средств телекоммуникации.

По дисциплине могут проводиться дополнительные занятия, в т.ч. в форме вебинаров. Расписание вебинаров и записи вебинаров публикуются в электронном курсе по дисциплине.

9.3. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 9.2.

Таблица 9.2 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения
С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)

Согласована на портале № 65300

С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами, определяющимися исходя из состояния обучающегося на момент проверки

9.4. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

– в печатной форме;
– в печатной форме с увеличенным шрифтом;
– в форме электронного документа;
– методом чтения ассистентом задания вслух;
– предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

– письменно на бумаге;
– набор ответов на компьютере;
– набор ответов с использованием услуг ассистента;
– представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:
– в форме электронного документа;
– в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:
– в форме электронного документа;
– в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:
– в форме электронного документа;
– в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 65300

ЛИСТ СОГЛАСОВАНИЯ

Рассмотрена и одобрена на заседании кафедры АОИ
протокол № 6 от «10» 12 2020 г.

СОГЛАСОВАНО:

Должность	Инициалы, фамилия	Подпись
Заведующий выпускающей каф. АОИ	А.А. Сидоров	Согласовано, d4895b45-5d88-49f8-b7c7-e8bf0196776a
Заведующий обеспечивающей каф. АОИ	А.А. Сидоров	Согласовано, d4895b45-5d88-49f8-b7c7-e8bf0196776a
Декан ФДО	И.П. Черкашина	Согласовано, 4580bdea-d7a1-4d22-bda1-21376d739cfc

ЭКСПЕРТЫ:

Доцент, каф. АОИ	Ю.В. Морозова	Согласовано, 8461038d-613f-4932-8e22-2b7293a14b92
Заведующий кафедрой, каф. АОИ	А.А. Сидоров	Согласовано, d4895b45-5d88-49f8-b7c7-e8bf0196776a

РАЗРАБОТАНО:

Доцент, каф. АОИ	Ю.В. Морозова	Разработано, 8461038d-613f-4932-8e22-2b7293a14b92
Доцент, каф. АОИ	Н.В. Пермякова	Разработано, 81211814-3a25-4c90-ad31-d4043108e403

Согласована на портале № 65300