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
«___» _____ 20__ 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: **ФДО, Факультет дистанционного обучения**
Кафедра: **АОИ, Кафедра автоматизации обработки информации**
Курс: **1, 2**
Семестр: **1, 2, 3**
Учебный план набора 2020 года

Распределение рабочего времени

№	Виды учебной деятельности	1 семестр	2 семестр	3 семестр	Всего	Единицы
1	Самостоятельная работа под руководством преподавателя	4	4	6	14	часов
2	Лабораторные работы	8	8	12	28	часов
3	Часы на контрольные работы	2	2	2	6	часов
4	Самостоятельная работа	49	49	79	177	часов
5	Всего (без экзамена)	63	63	99	225	часов
6	Подготовка и сдача экзамена	9	9	9	27	часов
7	Общая трудоемкость	72	72	108	252	часов
					7.0	З.Е.

Контрольные работы: 1 семестр - 1; 2 семестр - 1; 3 семестр - 1
Экзамен: 1, 2, 3 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Сенченко П.В.
Должность: Проректор по УР
Дата подписания: 18.12.2019
Уникальный программный ключ:
a1119608-cdff-4455-b54e-5235117c185c

Томск

Согласована на портале № 52537

Рассмотрена и одобрена на заседании кафедры
протокол № 332 от «10» 12 2019 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 38.03.05 Бизнес-информатика, утвержденного 11.08.2016 года, рассмотрена и одобрена на заседании кафедры АОИ «___» _____ 20__ года, протокол №_____.

Разработчики:

доцент каф. АОИ	_____	Н. В. Пермякова
доцент каф. АОИ	_____	Ю. В. Морозова

Заведующий обеспечивающей каф. АОИ	_____	А. А. Сидоров

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФДО	_____	И. П. Черкашина
Заведующий выпускающей каф. АОИ	_____	А. А. Сидоров

Эксперты:

Доцент кафедры автоматизации обработки информации (АОИ)	_____	А. А. Сидоров
Старший преподаватель кафедры технологий электронного обучения (ТЭО)	_____	А. В. Гураков

Согласована на портале № 52537

1. Цели и задачи дисциплины

1.1. Цели дисциплины

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

– 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

Дисциплина «Информатика и программирование» (Б1.Б.03.01) относится к блоку 1 (базовая часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Информатика и программирование.

Последующими дисциплинами являются: Тестирование программного обеспечения, Технологии программирования, Информатика и программирование.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– ОПК-1 способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности ;

– ОПК-3 способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях ;

В результате изучения дисциплины обучающийся должен:

– **знать** основные факты, концепции, принципы и теории, связанные с информатикой; типы систем счисления; графические способы представления алгоритмов; основные принципы структурного программирования; синтаксис языка программирования Си; методы обработки и способы реализации основных структур данных в объектно-ориентированных программных средах.

– **уметь** работать с современными средами программирования; представлять алгоритмы с помощью блок-диаграмм, диаграмм Насси-Шнайдермана, псевдокода и в виде программ на языке Си/Си++; разрабатывать объектно-ориентированные программы в современных инструментальных средах.

– **владеть** синтаксисом и правилами языка программирования Си; практическими навыками алгоритмизации поставленных перед ним задач; практическими навыками разработки и отладки программ на языке Си; практическими приемами объектно-ориентированного программирования; навыками работы в средах объектно-ориентированного программирования (составление, отладка и тестирование программ).

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 7.0 зачетных единицы и представлена в таблице 4.1.

Согласована на портале № 52537

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры		
		1 семестр	2 семестр	3 семестр
Контактная работа (всего)	42	12	12	18
Самостоятельная работа под руководством преподавателя (СРП)	14	4	4	6
Лабораторные работы	28	8	8	12
Часы на контрольные работы (всего)	6	2	2	2
Самостоятельная работа (всего)	177	49	49	79
Подготовка к контрольным работам	26	8	8	10
Оформление отчетов по лабораторным работам	32	8	12	12
Самостоятельное изучение тем (вопросов) теоретической части курса	119	33	29	57
Всего (без экзамена)	225	63	63	99
Подготовка и сдача экзамена	27	9	9	9
Общая трудоемкость, ч	252	72	72	108
Зачетные Единицы	7.0			

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

Названия разделов дисциплины	СРП, ч	Лаб. раб., ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
1 семестр					
1 Основы алгоритмизации	1	0	8	9	ОПК-1, ОПК-3
2 Интегрированная среда программирования DEV-CPP	1	0	10	11	ОПК-1, ОПК-3
3 Синтаксис и алфавит языка Си. Типы данных языка Си	1	0	10	11	ОПК-1, ОПК-3
4 Подготовка и исполнение программы на языке Си. Конструкции структурного программирования в Си	1	8	21	30	ОПК-1, ОПК-3
Итого за семестр	4	8	49	63	
2 семестр					
5 Функции	1	4	13	18	ОПК-1, ОПК-3
6 Массивы	1	4	16	21	ОПК-1, ОПК-3
7 Файлы в Си	1	0	10	11	ОПК-1, ОПК-3
8 Управление выводом в консоль	1	0	10	11	ОПК-1, ОПК-3
Итого за семестр	4	8	49	63	

Согласована на портале № 52537

3 семестр					
9 Введение в объектно-ориентированное программирование	1	0	10	11	ОПК-1, ОПК-3
10 Введение в Си++	1	0	11	12	ОПК-1, ОПК-3
11 Классы и объекты	2	8	26	36	ОПК-1, ОПК-3
12 Наследование и полиморфизм	1	4	22	27	ОПК-1, ОПК-3
13 Потоки ввода-вывода	1	0	10	11	ОПК-1, ОПК-3
Итого за семестр	6	12	79	99	
Итого	14	28	177	225	

5.2. Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)

Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя)

Названия разделов	Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)	Трудоемкость, ч	Формируемые компетенции
1 семестр			
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-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. Многофайловая компиляция. Сообщения об ошибках	1	ОПК-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-1, ОПК-3
	Итого	1	
4 Подготовка и	Этапы подготовки программы к исполне-	1	ОПК-1, ОПК-3

Согласована на портале № 52537

исполнение программы на языке Си. Конструкции структурного программирования в Си	нию. 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		
	Итого	1	
Итого за семестр		4	
2 семестр			
5 Функции	Синтаксис. Объявление и вызов функций. Локальные переменные. Выход из функций. Передача параметров по ссылке. Рекурсивные функции	1	ОПК-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-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-1, ОПК-3
	Итого	1	
8 Управление выводом в консоль	Win32 API. Типы данных Windows. Функции WinAPI	1	ОПК-1, ОПК-3
	Итого	1	

Согласована на портале № 52537

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-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-1, ОПК-3
	Итого	1	
11 Классы и объекты	Конструкторы. Деструкторы. Указатель this. Статические элементы класса. Перегрузка операций. Дружественные функции	2	ОПК-1, ОПК-3
	Итого	2	
12 Наследование и полиморфизм	Базовые и производные классы. Полиморфизм. Абстрактные классы. Типы наследования	1	ОПК-1, ОПК-3
	Итого	1	
13 Потоки ввода-вывода	Иерархия потоковых классов Си++. Консольный ввод-вывод. Форматирование вывода. Файловый ввод вывод	1	ОПК-1, ОПК-3
	Итого	1	
Итого за семестр		6	
Итого		14	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин												
	1	2	3	4	5	6	7	8	9	10	11	12	13
Предшествующие дисциплины													
1 Информатика и программирование	+	+	+	+	+	+	+	+	+	+	+	+	+
Последующие дисциплины													
1 Тестирование программного обеспечения	+		+	+	+	+	+	+	+	+	+	+	+
2 Технологии программирования	+								+		+	+	

Согласована на портале № 52537

3 Информатика и программирование					+	+	+	+	+	+	+	+	+

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Компетенции	Виды занятий			Формы контроля
	СРП	Лаб. раб.	Сам. раб.	
ОПК-1	+	+	+	Контрольная работа, Экзамен, Проверка контрольных работ, Отчет по лабораторной работе, Тест
ОПК-3	+	+	+	Контрольная работа, Экзамен, Проверка контрольных работ, Отчет по лабораторной работе, Тест

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Таблица 7.1 – Наименование лабораторных работ

Названия разделов	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-1, ОПК-3
	Обработка массивов	4	
	Итого	8	
Итого за семестр		8	
2 семестр			
5 Функции	Обработка матриц. Функции	4	ОПК-1, ОПК-3
	Итого	4	
6 Массивы	Работа со строками	4	ОПК-1, ОПК-3
	Итого	4	
Итого за семестр		8	
3 семестр			
11 Классы и объекты	Реализация класса.	4	ОПК-1, ОПК-3
	Конструкторы и деструкторы	4	
	Итого	8	
12 Наследование и полиморфизм	Перегрузка операторов.	4	ОПК-1, ОПК-3
	Итого	4	

Согласована на портале № 52537

Итого за семестр		12	
Итого		28	

8. Часы на контрольные работы

Часы на контрольные работы приведены в таблице 8.1.

Таблица 8.1 – Часы на контрольные работы

№	Вид контрольной работы	Трудоемкость, ч	Формируемые компетенции
1 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ОПК-1, ОПК-3
2 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ОПК-1, ОПК-3
3 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ОПК-1, ОПК-3
Итого		6	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
1 семестр				
1 Основы алгоритмизации	Самостоятельное изучение тем (вопросов) теоретической части курса	6	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	8		
2 Интегрированная среда программирования DEV-CPP	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	10		
3 Синтаксис и алфавит языка Си. Типы данных языка Си	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	10		
4 Подготовка и исполнение программы на	Самостоятельное изучение тем (вопросов) теоретической части курса	11	ОПК-1, ОПК-3	Контрольная работа, Отчет по лабораторной работе,

Согласована на портале № 52537

языке Си. Конструкции структурного программирования в Си	Оформление отчетов по лабораторным работам	8		Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	21		
	Выполнение контрольной работы	2	ОПК-1, ОПК-3	Контрольная работа
Итого за семестр		49		
	Подготовка и сдача экзамена	9		Экзамен
2 семестр				
5 Функции	Самостоятельное изучение тем (вопросов) теоретической части курса	7	ОПК-1, ОПК-3	Контрольная работа, Отчет по лабораторной работе, Тест, Экзамен
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	2		
	Итого	13		
6 Массивы	Самостоятельное изучение тем (вопросов) теоретической части курса	6	ОПК-1, ОПК-3	Контрольная работа, Отчет по лабораторной работе, Тест, Экзамен
	Оформление отчетов по лабораторным работам	8		
	Подготовка к контрольным работам	2		
	Итого	16		
7 Файлы в Си	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	10		
8 Управление выводом в консоль	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	10		
	Выполнение контрольной работы	2	ОПК-1, ОПК-3	Контрольная работа
Итого за семестр		49		
	Подготовка и сдача экзамена	9		Экзамен

Согласована на портале № 52537

3 семестр				
9 Введение в объектно-ориентированное программирование	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	10		
10 Введение в Си++	Самостоятельное изучение тем (вопросов) теоретической части курса	9	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	11		
11 Классы и объекты	Самостоятельное изучение тем (вопросов) теоретической части курса	16	ОПК-1, ОПК-3	Контрольная работа, Отчет по лабораторной работе, Тест, Экзамен
	Оформление отчетов по лабораторным работам	8		
	Подготовка к контрольным работам	2		
	Итого	26		
12 Наследование и полиморфизм	Самостоятельное изучение тем (вопросов) теоретической части курса	16	ОПК-1, ОПК-3	Контрольная работа, Отчет по лабораторной работе, Тест, Экзамен
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	2		
	Итого	22		
13 Потоки ввода-вывода	Самостоятельное изучение тем (вопросов) теоретической части курса	8	ОПК-1, ОПК-3	Контрольная работа, Тест, Экзамен
	Подготовка к контрольным работам	2		
	Итого	10		
	Выполнение контрольной работы	2	ОПК-1, ОПК-3	Контрольная работа
Итого за семестр		79		
	Подготовка и сдача экзамена	9		Экзамен
Итого		204		

Согласована на портале № 52537

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

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется.

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Пермякова Н.В. Информатика и программирование [Электронный ресурс]: учебное пособие.Томск : ФДО, ТУСУР, 2016. Доступ из личного кабинета студентов. — Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 28.09.2021).

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. Доступ из личного кабинета студента. — Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 28.09.2021).

12.2. Дополнительная литература

1. Подбельский, В.В. Курс программирования на языке Си [Электронный ресурс] [Электронный ресурс]: учебник / В.В. Подбельский, С.С. Фомин. — Электрон. дан. — Москва : ДМК Пресс, 2012. — 384 с. Доступ из личного кабинета студента. — Режим доступа: https://e.lanbook.com/book/4148 (дата обращения: 28.09.2021).

2. Ашарина, И.В. Объектно-ориентированное программирование в C++ [Электронный ресурс]: лекции и упражнения [Электронный ресурс] учебное пособие / И.В. Ашарина. — Электрон. дан. — Москва Горячая линия-Телеком, 2012. — 320 с. Доступ из личного кабинета. — Режим доступа: https://e.lanbook.com/book/5115 (дата обращения: 28.09.2021).

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Пермякова Н. В. Информатика и программирование : электронный курс / Н.В. Пермякова. – Томск : ФДО ТУСУР, 2016. Доступ из личного кабинета студента.

2. Пермякова Н. В. Информатика и программирование [Электронный ресурс]: методические указания по выполнению лабораторных работ. – Томск : ФДО ТУСУР, 2016. Доступ из личного кабинета студента. — Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 28.09.2021).

3. 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. Доступ из личного кабинета студента. — Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 28.09.2021).

4. Пермякова Н.В. Информатика и программирование [Электронный ресурс]: лабораторный практикум. -Томск : ФДО ТУСУР, 2016. Доступ из личного кабинета студента. — Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 28.09.2021).

5. Пермякова Н. В. Программирование C++: электронный курс / Н.В. Пермякова. – Томск : ФДО ТУСУР, 2014. Доступ из личного кабинета студента.

12.3.2. 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;

Согласована на портале № 52537

– в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:

– в форме электронного документа;

– в печатной форме.

12.4. Профессиональные базы данных и информационные справочные системы

1. ЭБС «Лань»: www.e.lanbook.com (доступ из личного кабинета студента).
2. eLIBRARY.RU: www.elibrary.ru
3. ЭБС «Юрайт»: https://urait.ru (доступ из личного кабинета студента)

13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

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

13.1.1. Материально-техническое и программное обеспечение дисциплины

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

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

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– 7-zip (с возможностью удаленного доступа)

– DEV C++ (с возможностью удаленного доступа)

– Google Chrome

– Kaspersky Endpoint Security 10 для Windows (с возможностью удаленного доступа)

– MS Office версий 2010 (с возможностью удаленного доступа)

– Microsoft Windows

– Notepad++ (с возможностью удаленного доступа)

– OpenOffice (с возможностью удаленного доступа)

13.1.2. Материально-техническое и программное обеспечение для лабораторных работ

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

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

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

– 7-zip (с возможностью удаленного доступа)

– DEV C++ (с возможностью удаленного доступа)

– Google Chrome

– Kaspersky Endpoint Security 10 для Windows (с возможностью удаленного доступа)

Согласована на портале № 52537

– MS Office версий 2010 (с возможностью удаленного доступа)
– Microsoft Windows
– OpenOffice (с возможностью удаленного доступа)

13.1.3. 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, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:

- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:

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

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

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 зрениями** 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.

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

1. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования. Какой тип передачи управления может использовать такая программа?

безусловный

Согласована на портале № 52537

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?
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;

Согласована на портале № 52537

```
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);
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;
```

Согласована на портале № 52537

```
}
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");
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++)
```

Согласована на портале № 52537

```
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

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");
```

Согласована на портале № 52537

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
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

Согласована на портале № 52537

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();
```

14.1.2. Экзамен

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

Вопрос №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.

Вопрос №2

Согласована на портале № 52537

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.
Вопрос №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.
Вопрос №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]
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.
Вопрос №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

Согласована на портале № 52537

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

Вопрос № 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.

Вопрос № 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..

Вопрос №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.

Согласована на портале № 52537

4. Алгоритм находит индекс последнего максимального элемента.

Вопрос №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.

Вопрос № 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.

Вопрос № 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.

Вопрос № 12.

Выберите верные утверждения:

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

Вопрос № 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.

Вопрос №14.

Выберите верные утверждения:

1. В последовательности int m; float b[12]; char* x; 13 лексем.

Согласована на портале № 52537

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.

Вопрос №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.

Вопрос № 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.

Вопрос № 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.

Вопрос №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
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
```

Согласована на портале № 52537

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
Вопрос 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
Вопрос № 20
Выберите верно записанные идентификаторы:
1. sinus
2. 2Summa
3. Sinu*s
4. double

14.1.3. Темы контрольных работ

Информатика и программирование
Вопрос № 1
Что будет выведено на экран при выполнении следующего фрагмента программы

Согласована на портале № 52537

```
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

Вопрос № 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

Вопрос № 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;`

Вопрос № 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.

Вопрос №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");
```

Согласована на портале № 52537

```
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.

Вопрос №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.

Вопрос №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.

Вопрос №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;
for(i=f;i<l;i+=z) S+=x[i];
return S; }
int main(int argc, char *argv[])
{ int y[10],i,p;
```

Согласована на портале № 52537

```
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.

Вопрос №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.

Вопрос №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;
}
```

1. 10

Согласована на портале № 52537

2. 8
3. 5
4. 6

14.1.4. Темы лабораторных работ

Проверка условий
Обработка матриц. Функции
Работа со строками
Реализация класса.
Конструкторы и деструкторы
Перегрузка операторов.
Обработка массивов

14.1.5. 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 необходимо осуществлять медленно, выделяя основные идеи; на основании изученного составить тезисы. Освоив материал, попытаться соотнести теорию с примерами из практики;
- если в тексте встречаются термины, следует выяснить их значение для понимания дальнейшего материала;
- необходимо осмысливать прочитанное и изученное, отвечать на предложенные вопросы.

Студенты могут получать индивидуальные консультации с использованием средств телекоммуникации.

По дисциплине могут проводиться дополнительные занятия в форме вебинаров. Расписание вебинаров публикуется в кабинете студента на сайте Университета. Запись вебинара публикуется в электронном курсе по дисциплине.

14.2. 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 14.

Таблица 14 – 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)
С нарушениями	Решение дистанционных тестов,	Преимущественно дистанционными

Согласована на портале № 52537

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

14.3. 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 № 52537