

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Воронежский государственный технический университет»

УТВЕРЖДАЮ

Декан факультета экономики, менеджмента и  
информационных технологий



/ Баркалов С.А./

31 августа 2021 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
«Основы алгоритмизации и структур данных»**

**Направление подготовки** 09.03.03 Прикладная информатика

**Профиль** Прикладная информатика в экономике цифрового общества

**Квалификация выпускника** бакалавр

**Нормативный период обучения** 4 года / 4 года и 11 м.

**Форма обучения** очная / заочная

**Год начала подготовки** 2021

Автор программы

Н.Г.Аснина

Заведующий кафедрой  
Систем управления и  
информационных  
технологий в  
строительстве

Е.Н.Десятирикова

Руководитель ОПОП

Н.Г.Аснина

Воронеж 2021

## 1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

### 1.1. Цели дисциплины

Целью данной дисциплины является формирование у студентов знаний основных приемов программирования, умений рационально выбирать алгоритмы и структуры данных для решения прикладных задач и владение технологией процедурного программирования.

### 1.2. Задачи освоения дисциплины

Задачами преподавания дисциплины являются:

- знакомство с современными методами программирования, основными структурами данных и алгоритмами обработки данных;
- получение навыков программирования на языке высокого уровне с использованием сложных структур данных и стандартных алгоритмов поиска, сортировки и других операций обработки;
- приобретение опыта работы с инструментальными средствами разработки программного обеспечения

## 2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Основы алгоритмизации и структур данных» относится к дисциплинам обязательной части блока Б1.

## 3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Основы алгоритмизации и структур данных» направлен на формирование следующих компетенций:

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

ОПК-4 - Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью;

ОПК-6 - Способен анализировать и разрабатывать организационно-технические и экономические процессы с применением методов системного анализа и математического моделирования;

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-2	знать современные средства разработки и анализа программного обеспечения на языках высокого уровня
	уметь работать с инструментальными средами разработки программного обеспечения
	владеет навыками работы с инструментальной средой создания и отладки программного кода Microsoft Visual Studio
ОПК-4	знать основные правила и стили оформления программ
	уметь применять стандарты построения исходного кода при разработке программы
	владеет навыками оформления программ на языке C#
ОПК-6	знать основные структуры данных в языках программирования, методы программирования и разработки

	эффективных алгоритмов решения прикладных задач
	уметь выбирать рациональные структуры данных и алгоритмы их обработки, обеспечивающие эффективную программную реализацию;
	владеет навыками составления, тестирования и отладки программы на языке C#

#### 4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Основы алгоритмизации и структур данных» составляет 6 з.е.

Распределение трудоемкости дисциплины по видам занятий  
**очная форма обучения**

Виды учебной работы	Всего часов	Семестры	
		2	3
<b>Аудиторные занятия (всего)</b>	84	48	36
В том числе:			
Лекции	34	16	18
Лабораторные работы (ЛР)	50	32	18
<b>Самостоятельная работа</b>	105	96	9
Часы на контроль	27	-	27
Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость:			
академические часы	216	144	72
зач.ед.	6	4	2

**заочная форма обучения**

Виды учебной работы	Всего часов	Семестры	
		2	3
<b>Аудиторные занятия (всего)</b>	16	6	10
В том числе:			
Лекции	6	2	4
Лабораторные работы (ЛР)	10	4	6
<b>Самостоятельная работа</b>	187	98	89
Часы на контроль	13	4	9
Виды промежуточной аттестации - экзамен, зачет	+	+	+
Общая трудоемкость:			
академические часы	216	108	108
зач.ед.	6	3	3

#### 5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

**5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий**

**очная форма обучения**

№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего
-------	-------------------	--------------------	------	-----------	-----	-------

						с час
1	Базовые понятия программирования и алгоритмизации	Алгоритм и программа как описание процесса обработки данных: История программирования. Понятие алгоритма, его исполнители и свойства, способы представления алгоритмов. Общее представление о структурах данных: основные понятия и свойства структур данных, роль и место структуры данных в программе. Понятие языка программирования: алфавит, синтаксис и семантика. Трансляция программ. Средства разработки ПО. Настройка среды разработки. Создание и управление программным проектом в среде разработки. Использование внешних ресурсов и стандартных библиотек для написания программ. Редактирование и отладка программ. Стили и правила оформления программного кода. Использование отладчика, оптимизатора	6	10	16	32
2	Основные конструкции программирования	Грамматика языка: Алфавит. Идентификаторы и операторы. Имена и объявления. Основные типы данных. Выражения и присваивание. Арифметические операции для целых и вещественных операндов. Управляющие конструкции: Операторы ветвления. Логические выражения. Вложение и множественный выбор. Операторы цикла. Цикл с предусловием и постусловием.	6	8	18	32
3	Структурный подход к программированию	Функции и передача параметров: Понятие функции и процедуры. Обзор стандартных функций. Рациональный размер и количество параметров. Аргументы и параметры. Передача параметров по ссылке и значению. Преобразование аргументов в точку вызова. Основные структуры данных: Прimitивные типы, массивы, записи, строки и операции над ними. Указатели и ссылки. Динамическое управление памятью. Понятие набора данных и файла. Открытие и закрытие потоков. Управляющая строка, форматы. Структурная декомпозиция: Область действия описаний. Переобъявления во вложенных блоках. Определения и объявления на внешнем уровне и внутреннем. Макросы с параметрами. Директивы условной компиляции. Make-файл. Создание собственных библиотек	6	8	18	32
4	Статические структуры данных	Простые структуры данных. Алгоритмы обработки простых структур данных. Линейные сложные и составные структуры данных. Массивы, записи, структуры, объединения. Основные понятия. Динамическое распределение памяти блоками фиксированного и переменного размеров при выполнении программ. Динамические переменные. Объявление и инициализация указателей. Обращение к переменной с помощью указателя. Косвенная адресация.	6	8	18	32
5	Алгоритмы обработки данных линейной структуры	Методы поиска. Последовательный поиск. Бинарный поиск. Интерполирующий поиск. Анализ сложности и эффективности алгоритмов. Сортировка выбором. Сортировка обменом (пузырек). Сортировка вставками. Сортировка с помощью прямого включения и обмена. Сортировка Шелла. Быстрая сортировка. Анализ сложности алгоритмов. Оценка эффективности методов сортировки	6	8	18	32
6	Динамические структуры данных	Динамические массивы записей структурного типа. Понятие рекурсии. Сравнение итерационных и рекурсивных алгоритмов. Односвязный линейный список как рекурсивный тип данных. Циклические списки, Двусвязный линейный список. Построение связанного списка на массиве и выполнение операций с ним. Сортировки на списках. Сортировка слиянием. Стеки и очереди. Сравнение построения стректов и очередей на связанных списках и массивах. Хеш-таблицы. Методы реализации хеш-таблиц	4	8	17	29
<b>Итого</b>			<b>34</b>	<b>50</b>	<b>105</b>	<b>18</b>

заочная форма обучения						9
№ п/п	Наименование темы	Содержание раздела	Лекц	Лаб. зан.	СРС	Всего часов
1	Базовые понятия программирования и алгоритмизации	Алгоритм и программа как описание процесса обработки данных: История программирования. Понятие алгоритма, его исполнители и свойства, способы представления алгоритмов. Общее представление о структурах данных: основные понятия и свойства структур данных, роль и место структуры данных в программе. Понятие языка программирования: алфавит, синтаксис и семантика. Трансляция программ. Средства разработки ПО. Настройка среды разработки. Создание и управление программным проектом в среде разработки. Использование внешних ресурсов и стандартных библиотек для написания программ. Редактирование и отладка программ. Стили и правила оформления программного кода. Использование отладчика, оптимизатора	2	2	30	34
2	Основные конструкции программирования	Грамматика языка: Алфавит. Идентификаторы и операторы. Имена и объявления. Основные типы данных. Выражения и присваивание. Арифметические операции для целых и вещественных операндов. Управляющие конструкции: Операторы ветвления. Логические выражения. Вложение и множественный выбор. Операторы цикла. Цикл с предусловием и постусловием.	2	2	30	34
3	Структурный подход к программированию	Функции и передача параметров: Понятие функции и процедуры. Обзор стандартных функций. Рациональный размер и количество параметров. Аргументы и параметры. Передача параметров по ссылке и значению. Преобразование аргументов в точке вызова. Основные структуры данных: Примитивные типы, массивы, записи, строки и операции над ними. Указатели и ссылки. Динамическое управление памятью. Понятие набора данных и файла. Открытие и закрытие потоков. Управляющая строка, форматы. Структурная декомпозиция: Область действия описаний. Переобъявления во вложенных блоках Определения и объявления на внешнем уровне и внутреннем. Макросы с параметрами. Директивы условной компиляции. Makeфайл. Создание собственных библиотек	2	2	32	36
4	Статические структуры данных	Простые структуры данных. Алгоритмы обработки простых структур данных. Линейные сложные и составные структуры данных. Массивы, записи, структуры, объединения. Основные понятия. Динамическое распределение памяти блоками фиксированного и переменного размеров при выполнении программ. Динамические переменные. Объявление и инициализация указателей. Обращение к переменной с помощью указателя. Косвенная адресация.	-	2	32	34
5	Алгоритмы обработки данных линейной структуры	Методы поиска. Последовательный поиск. Бинарный поиск. Интерполирующий поиск. Анализ сложности и эффективности алгоритмов. Сортировка выбором. Сортировка обменом (пузырек). Сортировка вставками. Сортировка с помощью прямого включения и обмена. Сортировка Шелла. Быстрая сортировка. Анализ сложности алгоритмов. Оценка эффективности методов сортировки	-	2	32	34
6	Динамические структуры данных	Динамические массивы записей структурного типа. Понятие рекурсии. Сравнение итерационных и рекурсивных алгоритмов. Односвязный линейный список как рекурсивный тип данных. Циклические списки, Двусвязный линейный список. Построение связанного списка на массиве и выполнение операций	-	-	31	31

		с ним. Сортировки на списках. Сортировка слиянием. Стеки и очереди. Сравнение построения стреков и очередей на связанных списках и массивах. Хеш-таблицы Методы реализации хеш-таблиц				
<b>Итого</b>			<b>6</b>	<b>10</b>	<b>187</b>	<b>203</b>

## 5.2 Перечень лабораторных работ

- 1 Написание и компиляция консольного приложения в среде разработки MICROSOFT VISUAL STUDIO
- 2 наблюдение
- 2 Форматированный ввод/вывод
- 3 Особенности использование данных различного типа
- 4 Использование функций стандартной библиотеки языка C#
- 5 Реализация разветвляющихся алгоритмов
- 6 Многоальтернативная разработка
- 7 Организация циклического вычислительного процесса
- 8 Конструирование функций
- 9 Работа с массивами
- 10 Работа с указателями
- 11 Работа со строками
- 12 Передача массива в функцию
- 13 Динамические массивы
- 14 Передача функций по указателю
- 15 Структуры данных
- 16 Файловый ввод/вывод
- 17 Статический и динамический многомерный массив
- 18 Алгоритмы сортировки структур данных
- 19 Массив структур
- 20 Структуры данных списки. Односвязный список. Двусвязный Список
- 21 Структуры данных: стеки и очереди

## 6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

В соответствии с учебным планом освоение дисциплины не предусматривает выполнение курсового проекта (работы) или контрольной работы.

## 7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

**7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания**

### 7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие	Критерии оценивания	Аттестован	Не аттестован
-------------	--------------------------------------	---------------------	------------	---------------

	<b>сформированность компетенции</b>			
ОПК-2	знать современные средства разработки и анализа программного обеспечения на языках высокого уровня	Активное участие в устных опросах на занятиях, правильно отвечает на теоретические вопросы промежуточного контроля, отвечает на теоретические и практические вопросы при защите	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь работать с инструментальными средами разработки программного обеспечения	Решение стандартных и конкретных прикладных задач на лабораторных работах	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеет навыками работы с инструментальной средой создания и отладки программного кода Microsoft Visual Studio	Решение стандартных и конкретных прикладных задач на лабораторных работах	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-4	знать основные правила и стили оформления программ	Активное участие в устных опросах на занятиях, правильно отвечает на теоретические и практические вопросы промежуточного контроля	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь применять стандарты построения исходного кода при разработке программы	Документирование кода и генерация технической документации встроенными средствами С на лабораторных работах	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеет навыками оформления программ на языке С#	Решение стандартных и конкретных прикладных задач на лабораторных работах	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
ОПК-6	знать основные структуры данных в языках программирования, методы программирования и разработки эффективных алгоритмов решения прикладных задач	Активное участие в устных опросах на занятиях, правильно отвечает на теоретические и практические вопросы промежуточного контроля	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	уметь выбирать рациональные структуры данных и алгоритмы их обработки, обеспечивающие эффективную программную реализацию;	Решение стандартных и конкретных прикладных задач на лабораторных работах	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
	владеет навыками составления, тестирования и отладки программы на языке С#	Решение стандартных и конкретных прикладных задач на лабораторных работах	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

### **7.1.2 Этап промежуточного контроля знаний**

Результаты промежуточного контроля знаний оцениваются в 2, 3 семестре для очной формы обучения, 2, 3 семестре для заочной формы обучения по двух/четырёхбалльной системе:

«зачтено»

«не зачтено»

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Зачтено	Не зачтено
ОПК-2	знать современные средства разработки и анализа программного обеспечения на языках высокого уровня	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь работать с инструментальными средами разработки программного обеспечения	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеет навыками работы с инструментальной средой создания и отладки программного кода Microsoft Visual Studio	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-4	знать основные правила и стили оформления программ	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь применять стандарты построения исходного кода при разработке программы	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеет навыками оформления программ на языке C#	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-6	знать основные структуры данных в языках программирования, методы программирования и разработки эффективных алгоритмов решения прикладных задач	Тест	Выполнение теста на 70-100%	Выполнение менее 70%
	уметь выбирать рациональные структуры данных и алгоритмы их обработки, обеспечивающие эффективную программную реализацию;	Решение стандартных практических задач	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеет навыками составления, тестирования и отладки программы на языке C#	Решение прикладных задач в конкретной предметной области	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

ИЛИ

«отлично»;

«хорошо»;

«удовлетворительно»;

«неудовлетворительно».

Компетенция	Результаты обучения, характеризующие	Критерии оценивания	Отлично	Хорошо	Удовл.	Неудовл.
-------------	--------------------------------------	---------------------	---------	--------	--------	----------

	<b>сформированность компетенции</b>					
ОПК-2	знать современные средства разработки и анализа программного обеспечения на языках высокого уровня	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь работать с инструментальным и средами разработки программного обеспечения	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеет навыками работы с инструментальной средой создания и отладки программного кода Microsoft Visual Studio	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-4	знать основные правила и стили оформления программ	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь применять стандарты построения исходного кода при разработке программы	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеет навыками оформления программ на языке C#	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
ОПК-6	знать основные структуры данных в языках программирования, методы программирования и разработки эффективных алгоритмов решения прикладных задач	Тест	Выполнение теста на 90-100%	Выполнение теста на 80-90%	Выполнение теста на 70- 80%	В тесте менее 70% правильных ответов
	уметь выбирать рациональные структуры данных и алгоритмы их обработки, обеспечивающие эффективную программную реализацию;	Решение стандартных практических задач	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены
	владеет навыками составления, тестирования и отладки программы на языке C#	Решение прикладных задач в конкретной предметной области	Задачи решены в полном объеме и получены верные ответы	Продемонстрирован верный ход решения всех, но не получен верный ответ во всех задачах	Продемонстрирован верный ход решения в большинстве задач	Задачи не решены

## 7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

### 7.2.1 Примерный перечень заданий для подготовки к тестированию

1

Формализованное описание процесса обработки данных – это ...
<b>программа</b>
алгоритм
информационная среда
транслятор

2

Какой тип следует выбрать для объявления «отлично»
<b>Char s[20];</b>
Int A;
Text t;
Short* str;

3

Программное средство состоит из
<b>прогр ммы на носителе данных и документации</b>
набора программ
программ и данных, расположенных на носителе инф рмации
совокупности логически связанных программ

4

Способность программного средства безотказно выполнять определенные функции при заданных условиях называется
<b>надежность</b>
качество
с провождаемость
устойчивость

5

Укажите целый тип данных
<b>int</b>
float
char
double

6

Укажите оператор
<b>A=5;</b>
int a[5];
int a(int);
a(5);

Что выведет функция: <pre>main() {   int x;   x = - 3 + 4 * 5 - 6; printf("% \n",x); }</pre>
<b>11</b>
-29
-41
0

Что напечатает следующая программа ? <pre>#define PRINTX printf("%d\n",x) main() {   int x=2,y,z;    x *= 3 + 2; PRINTX; }</pre>
<b>10</b>
5
2
20

Что напечатает следующая программа ? <pre>#define PRINT(int) printf("int = %d\n",int)  main() {   int x,y,z;    x = 03; y = 02; z = 01;   PRINT( x   y &amp; z ); }</pre>
<b>3</b>
0
2

Поставьте в соответствие этап	измененного цикла разработки ПС и его результат
Этап внешнего описания	Фиксация требований к ПС
Этап конструирования ПС	Разработка архитектуры, структуры программ и их детальной спецификация
Этап кодирования	Создание текстов на языках программирования, их отладка с тестированием
Этап ат естанции	Оценка качества ПС

Что напечатает следующая программа ?

```
#define PRINT3(x,y,z) printf("x=%d\ty=%d\tz=%d\t\n", x, y, z)
main()
{
int x, y, z;

x = y = z = 1;
++x || ++y && ++z;
PRINT3(x, y, z);
}
```

**x=2 y=1 z=1**

x=1 y=1 z=1

x=0 y=3 z=1

x=2 y=2 z=2

12

doubled;  
Чему будет равно d = 100/3;

33

33,33333

33e-33

0 33e-33

13

В каком случае требуется явное преобразование типа?

```
main()
{
double d;
float f;
long l;
int i;

i = l = f = d = 100/3;
}
```

**d = 100/3;**

i = 100/3;

l = 100/3;

f = 100./3;

14

Какой тип у переменной value при корректном ее выводе функцией  
printf("%d\t",value)

**int**

char

float

double

15

Поставьте в соответствие

**Объявление функции**                      **int sport(void);**

**Инициализация переменной**            **char c='d';**

**Задание массива**                        **int a[5];**

<b>Возвращение значения функции</b>	<b>return 5;</b>
-------------------------------------	------------------

16

При определении типа Char text[50]= «язык программирования». Что выведет функция printf("%c ",text[2])
<b>x</b>
<b>e</b>
<b>\n</b>
<b>%c</b>

17

<p>Что выведет программа?</p> <pre>#include&lt;stdio.h&gt;  int main(void) { int a[5] = { 1, 2, 3, 4, 5 }; int *ptr = (int*)&amp;a + 1);  printf("%d %d\n", *(a + 1), *(ptr - 1));  return 0; }</pre>
<b>2 5</b>
<b>2 2</b>
<b>2 1</b>
<b>ошибку</b>

18

<b>Поставьте в соответствие</b>	
<b>floatb</b>	<b>15.23</b>
<b>intb</b>	<b>5</b>
<b>charb</b>	<b>'d'</b>
<b>doublef</b>	<b>1e-</b>

19

<pre>void foo(int[][3]); int main(void) { int a[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} }; foo(a); printf("%d\n", a[2][1]); return 0; } void foo(int b[][3]) { ++b; b[1][1] = 9; } Что выведет программа?</pre>
<b>9</b>
<b>8</b>
<b>1</b>
<b>7</b>

Выберите два основных принципа модульного программирования
<b>Декомпозиция</b>
<b>Абстрагирование</b>
Детализация и обобщение
Обобщение

Выберите статический тип данных
<b>Вектор</b>
Стек
Очередь
Список

<pre>int main(void) { int a, b, c, d; a = 3; b = 5; c = a, b; d = (a, b); printf("c=%d ", c); printf("d=%d\n", d); return 0; } Что выведет программа?</pre>
<b>c=3 d=3</b>
c=5 d=3
c=3 d=5
c=5 d=5

Поставьте в соответствие конструкцию структурного программирования и фрагмент кода	
<b>следование</b>	<b>a=5; b+=a; return;</b>
<b>разветвления</b>	<b>if (a==b) return k;</b>
<b>повторения</b>	<b>while(!a) b++;</b>
<b>вызов функции</b>	<b>print("hello world");</b>

### 7.2.2 Примерный перечень заданий для решения стандартных задач

1. Нарисовать блок-схему алгоритма вычисления расстояния между двумя точками, заданными координатами  $x$  и  $y$ . По заданной блок-схеме алгоритма написать программу.
2. Вывести все нечетные числа кратные трем в интервале от  $A$  до  $B$  (возможно  $A > B$  и  $B > A$ ) включительно.
3. Получить в порядке убывания все простые числа не превосходящие заданное числа  $X$ .
4. Требуется распечатать текстовый файл объемом  $N$  байт. Сколько листов бумаги надо приготовить, если выбран формат печати:  $M$  строк по  $K$  символов в строке? При оценке считать, что файл не содержит управляющих

символов. Примечание. 1 символ занимает 1 байт.

5. С клавиатуры вводятся два целых числа. Сравнить эти числа. Результат вывести в виде  $3 < 5$  или  $3 = 3$ , или  $3 > 2$ . Примечание функцию `printf()` для вывода результата разрешается использовать только один раз.

6. Дан номер года. Определить количество дней в этом году. Примечание. При решении задачи принять во внимание, что в современном (григорианском) календаре каждый год, номер которого делится на 4, является високосным, за исключением тех номеров, которые делятся на 100 и не делятся на 400.

7. Составить программу, которая генерирует 10 случайных чисел в заданном пользователем диапазоне (от А до В). Примечание. Используйте функцию `rand()` (которая возвращает псевдослучайное целое число в диапазоне от 0 до `RAND_MAX (sizeof(int)=32767)`).

8. Напечатать таблицу, выводящую ежемесячную выплату по кредиту в течение года, если задана процентная ставка и сумма кредита. Примечание. Кредит дается на 1 год, процентная ставка – % от всей суммы кредита, сумма + проценты по кредиту распределяются равномерно на 12 месяцев.

9. Из файла считать вещественные числа, сформировать одномерный массив для которого вычислить: А) сумму всех отрицательных элементов; Б) произведение всех положительных. Результат записать в файл в виде текста, содержащего вычисленные значения.

10. Из файла считать вещественные числа, сформировать одномерный массив для которого вычислить: А) сумму элементов с четными номерами; Б) максимальный элемент. Результат записать в файл в виде текста, содержащего вычисленные значения.

11. Элемент матрицы называется локальным минимумом, если он строго меньше всех имеющихся у него соседей. Подсчитать количество локальных минимумов заданной матрицы размером 10 на 10. Найти сумму модулей элементов, расположенных выше главной диагонали.

12. Коэффициенты системы линейных уравнений заданы в виде прямоугольной матрицы. С помощью допустимых преобразований привести систему к треугольному виду. Найти количество строк, среднее арифметическое элементов которых меньше заданной величины.

13. Уплотнить заданную матрицу, удаляя из нее строки и столбцы, заполненные нулями. Найти номер первой из строк, содержащих хотя бы один положительный элемент.

14. Осуществить циклический сдвиг элементов прямоугольной матрицы на *n* элементов вправо или вниз (в зависимости от введенного режима), *n* может быть больше количества элементов в строке или столбце.

15. Создать структуру с именем `Student` с полями: фамилия, имя, номер группы, успеваемость (массив из пяти элементов). Сформировать односвязный список. Затем напечатать этот список в отсортированном по выбранному полю порядке. Затем удалить из списка информацию о тех студентах, у которых имеется по крайней мере две «2». И снова напечатать список.

16. Создать структуру с именем `Sotrudnik` с полями: фамилия, имя, должность, год поступления на работу. Сформировать односвязный список.

Затем напечатать этот список в отсортированном по выбранному полю порядке. Затем удалить из списка информацию о тех сотрудниках, которые были приняты на работу позже заданного года. И снова напечатать список.

17. Создать структуру с именем Sportsmen с полями: фамилия, имя, вид спорта, результат (занятое место). Сформировать односвязный список. Затем напечатать этот список в отсортированном по выбранному полю порядке. Затем удалить из списка информацию о тех спортсменах, которые имеют худший результат в своем виде спорта. И снова напечатать список.

18. За один просмотр файла  $f$ , элементами которого являются целые числа, без использования дополнительных файлов переписать его элементы во второй файл так, чтобы первоначально в каждой строке были записаны отрицательные числа, затем числа, равные 0, а затем положительные числа.

19. Написать программу, которая из текстового файла, разбитого на строки, переписывает все содержимое в другой текстовый файл, перемещая заглавные гласные латинские буквы каждой строки с сохранением их порядка следования в конец этой строки.

20. Дана непустая последовательность слов (строка), между которыми стоит пробел, за последним словом – точка. Вывести с сохранением исходного взаимного порядка три группы слов: слова, не содержащие цифр; слова, содержащие ровно три цифры и слова, полностью состоящие из цифр. Элементы второй и третьей групп могут повторяться.

### **7.2.3 Примерный перечень заданий для решения прикладных задач**

1. Написать и отладить программу типа «Конвертор», предназначенную

- 1) для пересчета курса валют;
- 2) вычисления объема куба и площади его боковой поверхности по заданной длине ребра;
- 3) вычисления силы тяжести при падении с заданной высоты тела заданной массы;
- 4) вычисления силы тока по известным значениям напряжения и сопротивления электрической цепи;
- 5) расчета расстояния между населенными пунктами (при указанном масштабе);
- 6) стоимости поездки, исходя из известного расстояния и стоимости бензина;
- 7) вычисления дохода по вкладу;
- 8) пересчета заданного временного интервала в часы и минуты;
- 9) вычисления возраста по дате рождения.

2. Некоторый промежуток времени задан в секундах. Представить его в виде количества суток, часов, минут и секунд. Например, если задано число 1769404, то вывести надо 4 числа: 20 11 30 4 (1769404 секунд – это 20 суток, 11 часов, 30 минут и 4 секунды).

3. Дано трехзначное число. Найдите число, которое получится, если цифры исходного числа переписать в обратном порядке.

4. Программа, определяющая варианты сдачи купюрами разного достоинства.

5. Выяснить пройдет ли кирпич с ребрами  $a$ ,  $b$ ,  $c$  в прямоугольное отверстие со сторонами  $x$  и  $y$ .

6. Реализовать программу для игры «Быки и коровы»

7. Запись имеет вид: название вуза, число студентов, количество факультетов. Добавить в конец файла информацию о трех новых вузах и посчитать общее число студентов.

8. Запись имеет вид: название издания, газеты или журнала, стоимость одного экземпляра, количество экземпляров в год. Вывести на печать информацию о самом дешевом издании.

9. Запись имеет вид: марка видеомэганитофона, стоимость, количество. Напечатать информацию об имеющихся в продаже мэганитофонах. При покупке их количество соответственно уменьшается. Предусмотреть удаление информации о видеомэганитофонах, количество которых равно нулю.

10. В населенном пункте проживает  $N$  человек. О каждом из жителей известно: фамилия, возраст, пол. Ввести информацию о жителях данного пункта и посчитать количество женщин и мужчин, вывести информацию о среднем возрасте мужчин, напечатать фамилии тех, чей возраст ниже среднего.

11. Описать структуру с именем TOVAR, с полями: название товара, цена за единицу, количество единиц в партии. Сформировать односвязный список. Затем напечатать элементы списка по возрастанию, используя в качестве первичного ключа – название товара, Затем напечатать информацию о тех товарах, у которых цена за единицу меньше средней цены всех товаров.

12. Описать структуру с именем STUDENT, с полями: фамилия, имя, номер группы, успеваемость (массив из пяти элементов). Сформировать односвязный список. Затем напечатать элементы списка по возрастанию, используя в качестве первичного ключа – номер группы. Затем напечатать фамилии студентов, у которых средний балл выше, чем средний балл по всем студентам.

#### **7.2.4 Примерный перечень вопросов для подготовки к зачету**

Не предусмотрено учебным планом

#### **7.2.5 Примерный перечень вопросов для подготовки к экзамену**

1. Понятия алфавит, синтаксис и семантика языков программирования. Понятие лексема, идентификатор, ключевые слова и комментарии. Принципы назначения идентификаторов программных объектов. Привести примеры написания комментариев в языке Си.
2. Понятие переменной и константы. Правила преобразования типов переменных в выражении. Привести пример арифметического

- выражения с явным и неявным преобразованием типов. Привести примеры различных видов констант языка С.
3. Понятие типа данных. Правила преобразования типов. Привести пример арифметического выражения с явным и неявным преобразованием типов
  4. Понятие операция, переменная, оператор, разделитель. Арифметические и логические операции. Оператор присваивания. Приоритет операций. Привести пример объявления, инициализации и использования переменной в выражении на языке Си. Привести примеры различных видов операторов. Показать, как использование разделителя меняет логику программы
  5. Линейный алгоритм. Привести пример решения задачи на его основе (условие задачи псевдокод или словесный, блок схема и запись на алгоритмическом языке).
  6. Алгоритм ветвления. Понятие управляющей структуры. Операторы управления процессом выполнения программы. Условный оператор и условная операция. Логические операции и операции отношения. Привести пример решения задачи с двухальтернативным выбором (условие задачи псевдокод или словесный, блок схема и запись на алгоритмическом языке)
  7. Циклический алгоритм с пред и постусловием. Привести пример (условие задачи, псевдокод или словесный алгоритм, блок схема и запись на языке Си). Циклы. Разновидности языковых конструкция построения циклов на языке С
  8. Функции в языке С. Объявление, определение и вызов функций. Привести примеры использования математических функций стандартной библиотеки
  9. Механизм передачи параметров возврат значений Формальные и фактические параметры. Передача параметров по значению и по ссылке. Привести примеры на структуре данных. Назначение заголовочных файлов.
  10. Локальные и глобальные переменные. Принцип наименьших привилегий. Пространство имен. Понятие области видимости. Показать как ограничить и расширить область видимости переменной, функции.
  11. Указатели. Объявление и инициализация указателей. Операции с указателями и особенности их использования. Операторы для работы с указателями, операции над указателями, массивы и указатели, строки и указатели, преобразования типов в указателях для языка С. Привести примеры.
  12. Динамическая память. Функции для выделения и освобождения динамической памяти в языке С. Привести пример динамического создания массивов.
  13. Понятие потокового ввода-вывода. Стандартные потоки вывода и функции работы с ними. Стандартные потоки ввода и функции работы с ними

14. Файлы. Принципы работы с текстовыми файлами. Привести пример кода записи целого числа в файл и чтения строки из файла. Бинарные файлы. Преимущества и недостатки работы с бинарными файлами.
15. Структура программы на языке Си. Параметры функции main. Привести пример их использования. Директивы препроцессора, преимущества их использования.
16. Значение типизации в программировании. Фундаментальные типы данных языка Си их модификации. Привести обоснование выбора типа данных для различных задач. Оператор sizeof. Создание собственного типа данных. Привести примеры преимуществ использования.
17. Понятие структура данных, типы данных. Классификация структур данных. Понятие составной (агрегированной) тип данных. Перечисления: объявление, инициализация, примеры использования.
18. Сложные (агрегированные) типы данных, структуры данных языка. Разновидности встроенных в языке C сложных (составных) типов данных (массивы). Объявление, инициализация, преобразования и выполнение операций с элементами массивов.
19. Создание двумерных массивов с использованием указателей. Передача массивов в функцию и возврат двумерных массивов из функции
20. Многомерные статические массивы. Привести пример объявления, инициализации, обращения к элементу
21. Записи. Структура в языке C. Синтаксис, объявления, инициализация, принцип доступа к полям структуры. Привести примеры.
22. Массивы строк. Объявление, инициализация, обращение к отдельным строкам. Операции со строками
23. Массивы структур данных. Объявление, обращение к отдельной записи. Предложить реализацию поиска записи по заданному значению поля.
24. Понятие списка. Линейный список. Операции с линейным списком (создание элемента, добавление и удаление узла, поиск узла)
25. Понятие списка. Двусвязный список. Операции с двусвязным списком (создание элемента, добавление и удаление узла, поиск узла)
26. Понятие стека. Организация стека на массиве и связанном списке. Операции со стеком. .
27. Понятие очередь. Организация очереди. Операции с очередями. Привести пример
28. Понятие алгоритма, алгоритмизации. Свойства алгоритма.
29. Понятие рекурсии. Рекурсивный и итерационный алгоритмы. Привести пример решения одной задачи с использованием рекурсии и цикла.
30. Бинарный (двоичный) поиск. Алгоритм словесного описания, блок – схема, программный код поиска элемента у в массиве N из m элементов, отсортированного в возрастающем порядке.
31. Линейный (последовательный) поиск. Алгоритм словесного описания, блок – схема, составление функции нахождения минимального

элемента в массиве.

32. Интерполирующий поиск. Алгоритм словесного описания, блок – схема, программный код поиска элемента у в массиве N из m элементов, отсортированного в возрастающем порядке
33. Сортировка методом простого выбора. Алгоритм словесного описания, псевдокод, блок – схема, программный код по возрастанию элементов массива
34. Быстрая сортировка. Алгоритм словесного описания, псевдокод, блок – схема, программный код по возрастанию элементов массива
35. Сортировка методом простого обмена («пузырька»). Алгоритм словесного описания, блок – схема, программный код по возрастанию элементов массива
36. Шейкер - сортировка. Алгоритм словесного описания, блок – схема, программный код по возрастанию элементов массива.
- 37.. Сортировка вставками. Алгоритм словесного описания, блок – схема, реализация функции сортировки вставками по возрастанию элементов массива
38. Сортировка слиянием. Алгоритм словесного описания, блок – схема, программный код по возрастанию элементов массива
39. Сортировка Шелла. Алгоритм словесного описания, блок – схема, программный код по возрастанию элементов массива
40. Сортировка расческой. Алгоритм словесного описания, блок – схема, программный код по возрастанию элементов массива

#### **7.2.6. Методика выставления оценки при проведении промежуточной аттестации**

*Оценка складывается из результатов текущего балльно-рейтингового оценивания и экзамена.*

*Балльное-рейтинговое оценивание проводится по следующим критериям:*

*А) решение задач в ходе лабораторных работ и защита индивидуального домашнего задания (1 балл за каждую задачу до 3 баллов на каждую лабораторную работу, и суммарный балл (N) преобразуется в пятибалльную оценку по формуле  $N/60*5$ ).*

*Б) тестирование по материалам лекции (отношение числа правильно выполненных тестовых заданий от кол-ва полученных, оценивается как среднее значение по всем лекциями – максимум 1 балл)*

*Экзамен проводится по билетам, каждый из которых содержит 2 теоретических вопроса и практическую задачу. Каждый правильный*

ответ на теоретический вопрос оценивается в 0,5 балл, на подготовку отводится

15 минут. Практическое задание состоит в реализации программы по заданным требованиям. Оценка в 1 балл ставится по результатам проверки выполнения программы на контрольном примере каждой из трех подзадач. Время выполнения практического задания 90 минут.

Суммарная оценка формируется из результатов балльно-рейтингового оценивания и экзамена.

1. Оценка «Неудовлетворительно» ставится в случае, если студент набрал менее 3 баллов.
2. Оценка «Удовлетворительно» ставится в случае, если студент набрал не менее 3 баллов.
3. Оценка «Хорошо» ставится в случае, если студент набрал не менее 4 баллов.
4. Оценка «Отлично» ставится, если студент набрал от 5 баллов и выше.

### 7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Базовые понятия программирования и алгоритмизации	ОПК-2, ОПК-4, ОПК-6	Тест, устный опрос, защита лабораторных работ, защита курсового проекта
2	Основные конструкции программирования	ОПК-2, ОПК-4, ОПК-6	Тест, устный опрос, защита лабораторных работ, защита курсового проекта
3	Структурный подход к программированию	ОПК-2, ОПК-4, ОПК-6	Тест, устный опрос, защита лабораторных работ, защита курсового проекта
4	Статические структуры данных	ОПК-2, ОПК-4, ОПК-6	Тест, устный опрос, защита лабораторных работ, защита курсового проекта
5	Алгоритмы обработки данных линейной структуры	ОПК-2, ОПК-4, ОПК-6	Тест, устный опрос, защита лабораторных работ, защита курсового проекта
6	Динамические структуры данных	ОПК-2, ОПК-4, ОПК-6	Тест, устный опрос, защита лабораторных работ, защита курсового проекта

**7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методики выставления оценки при проведении промежуточной аттестации.

Решение стандартных задач осуществляется при помощи инструментальной среды Microsoft Visual Studio. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Решение прикладных задач осуществляется при помощи инструментальной среды Microsoft Visual Studio. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методики выставления оценки при проведении промежуточной аттестации.

Защита курсового проекта или отчета по всем видам практик осуществляется согласно требованиям, предъявляемым к работе, описанным в методических материалах. Примерное время защиты на одного студента составляет 20 мин.

## **8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ)**

### **8.1 Перечень учебной литературы, необходимой для освоения дисциплины**

1. Курипта, О.В. Основы программирования и алгоритмизации : практикум /О.В. Курипта, О.В. Минакова, Д.К. Проскурин. — Воронеж: Воронежский государственный архитектурно-строительный университет, 2015. — 133 с. — Режим доступа: <http://www.iprbookshop.ru/59123>

2. Орлов С.А. Теория и практика языков программирования [Текст] : учебник. - Москва ; Санкт-Петербург ; Нижний Новгород [и др.] : Питер, 2014 (Чехов : Первая Образцовая тип., фил. "Чеховский Печатный Двор", 2014). - 688 с.

3. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона [Электронный ресурс]/ Никлаус Вирт— Электрон. текстовые данные.— М.: ДМК Пресс, 2010.— 272 с.— Режим доступа: <http://www.iprbookshop.ru/7965>

4. Каширин И.Ю. От С к С++ [Электронный ресурс]: учебное пособие/ Каширин И.Ю., Новичков В.С.— Электрон. текстовые данные.— М.: Горячая линия - Телеком, 2012.— 334 с.— Режим доступа: <http://www.iprbookshop.ru/12022>.

5. Разработка программных приложений в среде Microsoft Visual C++ 2010 [Текст] : методические указания к выполнению лабораторных работ для студентов бакалавриата направления 09.03.02 "Информационные системы и

технологии" / Воронеж. гос. архитектур.-строит. ун-т ; [сост.: О. В. Минакова, О. В. Курипта]. - Воронеж: [б. и.], 2017 (Воронеж: Отдел оперативной полиграфии изд-ва учеб. лит. и учеб.-метод. пособий Воронежского ГАСУ, 2017). - 34 с.

б. Основы программирования и алгоритмизации. Методические указания к выполнению курсового проекта для студентов бакалавриата направления 09.03.02 "Информационные системы и технологии"/ Воронеж. гос. архитектур.-строит. ун-т ; [сост.: О. В. Минакова, О. В. Курипта]. - Воронеж: [б. и.], 2017 (Воронеж: Отдел оперативной полиграфии изд-ва учеб. лит. и учеб.-метод. пособий Воронежского ГАСУ, 2018). - 34 с.

**8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:**

*Программное обеспечение*

Инструментальная среда разработки ПО – MICROSOFT VISUAL STUDIO 2017 и выше.

Информационно-справочные системы

1. The Visual C++ programming language and development tools <https://docs.microsoft.com/>
  2. Описание языка программирования ANSI C <http://www.opennet.ru/docs/RUS/ansi-c/>
  3. Библиотека алгоритмов <http://algolist.manual.ru/>
- Авторский электронный образовательный ресурс дисциплины:
1. Практикум по Си. Основы программирования и алгоритмизации <https://sites.google.com/view/course-of-study1-c>
  2. Основы структурного программирования на языке Си <https://stepik.org/course/92532>
  3. Работа с массивами на C <https://stepik.org/course/103283>

## **9 МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

Компьютерный класс с предустановленной средой разработки приложений MICROSOFT VISUAL STUDIO 2010 и выше.

Мультимедийные средства: наборы файлов презентаций по темам лекционных занятий, комплект видеороликов по установке, настройке и примерам использования инструментальных средств технологии программирования.

Средства мониторинга – программа тестирования по модулям дисциплины с базами тестовых вопросов.

## **10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

По дисциплине «Основы алгоритмизации и структур данных» читаются лекции, проводятся лабораторные работы.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

Лабораторные работы выполняются на лабораторном оборудовании в соответствии с методиками, приведенными в указаниях к выполнению работ.

Вид учебных занятий	Деятельность студента
Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебника, проработать дополнительную литературу и источники, решить задачи и выполнить другие письменные задания.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоению учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: <ul style="list-style-type: none"><li>- работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций;</li><li>- выполнение домашних заданий и расчетов;</li><li>- работа над темами для самостоятельного изучения;</li><li>- участие в работе студенческих научных конференций, олимпиад;</li><li>- подготовка к промежуточной аттестации.</li></ul>
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед зачетом, экзаменом, экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

№ п/п	Перечень вносимых изменений	Дата внесения изменений	Подпись заведующего кафедрой, ответственной за реализацию ОПОП
----------	-----------------------------	----------------------------	--