



Федеральное агентство морского и речного транспорта
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Государственный университет морского и речного флота
имени адмирала С.О. Макарова»**

Воронежский филиал
**Федерального государственного бюджетного образовательного
учреждения высшего образования**
**«Государственный университет морского и речного флота
имени адмирала С.О. Макарова»**

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ
Б1.Б.15 «Технология программирования»
(Приложение к рабочей программе дисциплины)

Уровень образования:	Высшее образование – бакалавриат	
Направление подготовки:	09.03.02 Информационные системы и технологии	
Язык обучения:	Русский	
Кафедра:	Математики, информационных систем и технологий	
Форма обучения:	Очная	Заочная
Курс:	3	4
Составитель:	Плотников С.Н.	

ВОРОНЕЖ 2019 г.

СОДЕРЖАНИЕ

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ	3
1.1 Перечень компетенций и этапы их формирования в процессе освоения дисциплины.....	3
1.2 Паспорт фонда оценочных средств для проведения текущей и промежуточной аттестации обучающихся	5
1.3 Критерии оценивания результата обучения по дисциплине и шкала оценивания	8
2. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ	9
2.1 Текущий контроль	9
2.2 Критерии оценки качества освоения дисциплины.....	34
3. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ	36
3.1 Теоретические вопросы и практические задания для проведения экзамена.....	36
3.2 Показатели, критерии и шкала оценивания ответов на экзамене, подготовки курсовой работы.....	39

1. ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ

1.1 Перечень компетенций и этапы их формирования в процессе освоения дисциплины

В результате освоения ОПОП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Код компетенции	Содержание компетенции	Планируемые результаты освоения дисциплины
ОПК-5	способность использовать современные компьютерные технологии поиска информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению	<p>знать: основы информационной безопасности; основы поиска информации в компьютерных сетях; основные сведения о дискретных структурах, используемых в персональных компьютерах; основные алгоритмы типовых численных методов решения математических задач; один из языков программирования; структуру локальных и глобальных компьютерных сетей.</p> <p>уметь: работать в качестве пользователя персонального компьютера; использовать информацию компьютерных сетей в своей профессиональной деятельности для повышения мастерства; выполнять расчеты с применением современных технических средств; использовать внешние носители информации для обмена данными между машинами, создавать резервные копии, архивы данных и программ; использовать языки и системы программирования, работать с программными средствами общего назначения</p> <p>владеть: навыками систематизации информации; методами поиска и обмена информацией в компьютерных сетях; теоретическими и программными средствами защиты информации при работе с компьютерными системами, включая системы антивирусной защиты.</p>
ПК-15	способность участвовать в работах по доводке и освоению информационных технологий в ходе внедрения и эксплуатации информационных систем	<p>Знать: основы применения средств информационных технологий, при внедрении и эксплуатации информационных систем в учебной деятельности.</p> <p>Уметь: критически оценивать свои достоинства и недостатки, наметить пути и выбрать средства развития достоинств и устранения недостатков.</p> <p>Владеть: способностью развития познавательных интересов, интеллектуальных и творческих способностей средствами информационных технологий, воспитания ответственного отношения к информации с</p>

		учетом правовых и этических аспектов ее распространения; избирательного отношения к полученной информации.
ПК-17	<p>способность использовать технологии разработки объектов профессиональной деятельности, в областях: машиностроение, приборостроение, наука, техника, образование, медицина, административное управление, юриспруденция, бизнес, предпринимательство, коммерция, менеджмент, банковские системы, безопасность информационных систем, управление технологическими процессами, механика, техническая физика, энергетика, ядерная энергетика, силовая электроника, металлургия, строительство, транспорт, железнодорожный транспорт, связь, телекоммуникации, управление инфокоммуникациями, почтовая связь, химическая промышленность, сельское хозяйство, текстильная и легкая промышленность, пищевая промышленность, медицинские и биотехнологии, горное дело, обеспечение безопасности подземных предприятий и производств, геология, нефтегазовая отрасль, геодезия и картография, геоинформационные системы, лесной комплекс, химико-лесной комплекс, экология, сфера сервиса, системы массовой информации, дизайн,</p>	<p>Знать: теоретические основы использования технологии разработки объектов профессиональной деятельности в различных областях человеческой деятельности.</p> <p>Уметь: применять теоретические знания в области профессиональной деятельности, включающей исследование, разработку, внедрение и сопровождение информационных технологий и систем.</p> <p>Владеть: способностью использовать технологии разработки объектов профессиональной деятельности в различных областях человеческой деятельности.</p>

	медиаиндустрия, а также предприятия различного профиля и все виды деятельности в условиях экономики информационного общества	
ПК-34	способность к инсталляции, отладке программных и настройке технических средств для ввода информационных систем в опытную и промышленную эксплуатацию	<p>Знать: структуру программного и технического обеспечения, их основные функции и характеристики, методы инсталляции, отладку программных и настройку технических средств, механизмы администрирования, тенденции их развития (управление распределением памяти для объектов ИС, установление квот памяти для пользователей ИС, управления доступностью данных, включая режимы (состояния)).</p> <p>Уметь: выполнять процедуры настройки технических средств информационных систем.</p> <p>Владеть: средствами и средой программирования, современными технологиями программирования, методами настройки и отладки осуществления перехода от управления функционированием отдельных устройств к анализу трафика в отдельных участках сети.</p>

1.2 Паспорт фонда оценочных средств для проведения текущей и промежуточной аттестации обучающихся

№ п/п	Контролируемые темы дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Тема 1. Понятие алгоритма и алгоритмического языка	ОПК-5, ПК-15, ПК-17, ПК-34	Вопросы для контроля знаний, тестирование, экзамен
2	Тема 2 Данные и память. Абстракция данных.	ОПК-5, ПК-15, ПК-17, ПК-34	Вопросы для контроля знаний, тестирование, курсовая работа, экзамен
3	Тема 3 Типы данных, операции и операторы языка Паскаль	ОПК-5, ПК-15, ПК-17, ПК-34	Вопросы для контроля знаний, тестирование, курсовая работа, экзамен
4	Тема 4 Процедурная структура языка Паскаль	ОПК-5, ПК-15, ПК-17, ПК-34	Вопросы для контроля знаний, тестирование, курсовая работа, экзамен
5	Тема 5 Принцип модульности в программировании	ОПК-5, ПК-15,	Опрос на лабораторном практикуме,

		ПК-17, ПК-34	практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
6	Тема 6 Реализация модульности в языке Паскаль	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
7	Тема 7 Организация работы с файлами	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
8	Тема 8 Динамические структуры и управление памятью в языке Паскаль	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
9	Тема 9 Принцип объектно-ориентированного подхода к разработке программного обеспечения	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
10	Тема 10 Конструирование, тестирование, и отладка программ	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
11	Тема 11 Языки и системы программирования.	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний,

			тестирование, курсовая работа, экзамен
12	Тема 12 Операторы и выражения.	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
13	Тема 13 Управляющие операторы	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
14	Тема 14 Массивы и строки. Введение в классы, объекты и методы. Файлы и потоки	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
15	Тема 15 Виды классов Разработка Windows-приложений. ООП в деталях и примерах	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
16	Тема 16 Виды классов. ООП в деталях и примерах. Наследование. Разработка Windows-приложений	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
17	Тема 17 Интерфейсы и структуры.	ОПК-5, ПК-15, ПК-17, ПК-34	Опрос на лабораторном практикуме, практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
18	Тема 18 Делегаты, события и лямбда-выражения	ОПК-5, ПК-15,	Опрос на лабораторном практикуме,

		ПК-17, ПК-34	практические задания для самостоятельной работы, вопросы для контроля знаний, тестирование, курсовая работа, экзамен
--	--	-----------------	----------------------------------------------------------------------------------------------------------------------

Тематика лабораторных работ

№ п/п	Наименование лабораторных работ
1.	Способы записи алгоритмов. Составление блок-схем
2.	Использование стандартных типов, задание типов определяемых пользователем при разработке программ
3.	Разработка и реализация программ с использованием стандартных и определенных пользователем типов данных
4.	Разработка программ линейной, разветвляющейся, циклической структуры. Составление подпрограмм – процедур, подпрограмм – функций
5.	Реализация программ использующих основные структуры программирования
6.	Разработка программ с переменными типа запись. Работа с файлами данных
7.	Создание программ с переменными типа запись
8.	Манипулирование адресами области памяти с помощью указателей
9.	Разработка и реализация программ с файлами данных
10.	Динамическое программирование
11.	Разработка рекурсивных алгоритмов
12.	Реализация рекурсивных структур. Применение рекурсивных алгоритмов для различных сортировок
13.	Модульная структура программы. Отладка модулей
14.	Разработка и реализация модульных программ

1.3 Критерии оценивания результата обучения по дисциплине и шкала оценивания

Уровни сформированности компетенции	Основные признаки уровня
Неудовлетворительно	студент обнаруживает незнание ответа на соответствующее задание, допускает ошибки в формулировке определений и правил, искажающие их смысл, беспорядочно и неуверенно излагает материал; отмечаются такие недостатки в подготовке студента, которые являются серьезным препятствием к успешному овладению последующим материалом.
Пороговый (базовый) уровень (Оценка «3», Зачтено) (обязательный по отношению ко всем выпускникам к моменту завершения ими обучения по ОПОП)	ставится, если студент обнаруживает знание и понимание основных положений данного задания, но: 1) излагает материал неполно и допускает неточности в определении понятий или формулировке правил; 2) не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры; 3) излагает материал непоследовательно и допускает ошибки
Повышенный (продвинутый) уровень (Оценка «4», Зачтено)	студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2

(превосходит пороговый (базовый) уровень по одному или нескольким существенным признакам)	ошибки, которые сам же исправляет.
Высокий (превосходный) уровень (Оценка «5», Зачтено) (превосходит пороговый (базовый) уровень по всем существенным признакам, предполагает максимально возможную выраженность компетенции)	полно и аргументированно отвечает по содержанию задания; обнаруживает понимание материала, может отлично обосновать свои суждения, применить знания на практике, привести необходимые примеры не только по учебнику, но и самостоятельно составленные; излагает материал последовательно и правильно.

2. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ

2.1 Текущий контроль

Тема 1. Понятие алгоритма и алгоритмического языка

Контрольные вопросы:

- 1.Алгоритм, его свойства, способы представления.
- 2.Программы на языках высокого уровня.
- 3.Постановка задачи и спецификация программы.
- 4.Основные этапы решения задач на ЭВМ.

Вопросы для контроля знаний:

1. Критерии качества программ; диалоговые программы.
2. Жизненный цикл программ.
3. Каковы возможные подходы к определению понятия алгоритм?
4. Кто (что) может быть исполнителем алгоритма?
5. В чем особенности графического способа представления алгоритмов?
6. Каковы основные алгоритмические структуры?
7. Чем определяются свойства алгоритмов «дискретность», «определенность», «понятность», «результативность», «массовость»?
8. Что такое алгоритмический язык?

Тема 2 Данные и память. Абстракция данных.

Контрольные вопросы:

1. Определение типа данных.
2. Статический и динамический контроль типов и классификация языков в соответствии с ним.
3. Слабая, сильная и строгая типизации в статически типизируемых языках.

Вопросы для контроля знаний:

1. Конструкции структурного программирования и теорема о структурировании.
2. Абстракция управления.

Тема 3 Типы данных, операции и операторы языка Паскаль

Контрольные вопросы:

1. Введение в объектно-ориентированное программирование.
2. Объект, как активный процесс.
3. Основные понятия: инкапсуляция, наследование и полиморфизм.
4. Наследование и переопределение.

Вопросы для контроля знаний:

1. Раннее и позднее связывание.
2. Виртуальные методы.
3. Конструкторы и деструкторы.
4. Архитектура, управляемая событиями.

Тема 4 Процедурная структура языка Паскаль

Контрольные вопросы:

1. Лексическая структура языка. Структура паскаль- программ, разделы и их назначение.
2. Понятия типа, иерархия типа. Целые типы, литерный тип, вещественные типы. Генератор случайных чисел.
3. Булевский тип. Логические операции, логические функции.
4. Выражения, правила их записи и выполнения, классификация выражения по типу результата.
5. Ввод информации, вывод информации на экран. Форматный вывод.
6. Программирование ветвлений. Понятие цикла, циклы с заданным числом повторений и итерационные. Специальный выход из цикла.

Вопросы для контроля знаний:

1. Вложенные циклы, их организация с помощью всех трех операторов. Правила использования.
2. Перечислимый тип. Правила использования, действия с данными указанного типа
3. Ограниченный тип, области его применения. Функции low, high. Приведение типа переменной.
4. Одномерные, двумерные массивы, ввод/вывод, использование индексов нечислового типа. Ошибки в работе с регулярными типами, режимы компиляции. \$R+ и \$R-..
5. Строковый тип. Процедуры и функции для обработки строк. (пример программы обработки текста с использованием строк).
6. Множества, действия с множествами.
7. Понятие структуры данных запись. Иерархические записи, массивы записи, оператор присоединения. Записи с вариантами, действия с ними, размещение в памяти.

Тема 5 Принцип модульности в программировании

Контрольные вопросы:

1. Процедуры. Описание и вызов.
2. Формальные и фактические параметры, их взаимодействие.
3. Параметры-переменные, параметры-значения, параметры-константы.
4. Параметры сложных типов.
5. Бестиповые параметры. Открытые параметры и строки.
6. Функции. Описание, вызов.

Вопросы для контроля знаний:

1. Область действия имен.

2. Формальные и фактические параметры.
3. Локальные и глобальные идентификаторы.
4. Процедурный и функциональный типы. Правила использования.
5. Рекурсивные алгоритмы.
6. Порядок отведения памяти под локальные параметры. Пример программы.
7. Синтаксические анализаторы.
8. Элементы металингвистических формул. Пример программы.

Лабораторная работа № 1 Способы записи алгоритмов. Составление блок-схем

Цель работы: Научиться работать с алгоритмами и блок-схемами. Научиться составлять алгоритмы различных структур.

Вопросы по лабораторной работе 1

1. Что такое блок-схема?
2. Какие типы блоков бывают?
3. Какие блоки используются при реализации линейного, разветвляющегося, циклического алгоритмов?
4. Можно ли составить разные варианты блок-схем для одной и той же задачи?
5. Какие виды циклического алгоритма бывают?
6. Какие пункты должны присутствовать в любом цикле?
7. Что такое выполнение блок-схемы?
8. Для чего следует выполнять блок-схему?

Тема 6 Реализация модульности в языке программирования

Контрольные вопросы:

1. Абстракция модульности.
2. Модульное программирование.
3. Принцип скрытия информации.
4. Аксиома модульности.
5. Принцип сборочного программирования.
6. Определение модуля.
7. Модули в языках программирования.

Вопросы для контроля знаний:

1. Процедурная абстракция.
2. Основные характеристики программного модуля.
3. Гипотеза о глобальных данных.
4. Размер модуля.
5. Сцепление модулей.
6. Связность модуля.
7. Методы разработки структуры программы.

Лабораторная работа № 2 Использование стандартных типов, задание типов определяемых пользователем при разработке программ

Цель работы: Ознакомиться со структурой программы, научиться работать со всеми типами данных, операторами.

Вопросы по лабораторной работе 2

1. Определение типа данных.
3. Статический и динамический контроль типов и классификация языков в соответствии с ним.
2. Слабая, сильная и строгая типизации в статически типизируемых языках.
3. Конструкции структурного программирования и теорема о структурировании.
4. Абстракция управления.

Тема 7 Организация работы с файлами

Контрольные вопросы:

1. Структура модуля.
2. Использование модулей.
3. Пример с интерфейсной и реализующей частью.
4. Правила использования модулей.
5. Пример модуля с разделом инициализации.

Вопросы для контроля знаний:

1. Компиляция модулей.
2. Процедуры и функции модуля CRT.TPU.
3. Процедуры и функции модуля STRINGS.TPU.
4. Процедуры и функции модуля DOS.TPU.

Лабораторная работа № 3 Разработка и реализация программ с использованием стандартных и определенных пользователем типов данных

Цель работы: Приобретение навыков разработки программ с использованием процедур и функций пользователя

Вопросы по лабораторной работе 3

1. Что называется подпрограммой?
2. Какова структура программы с подпрограммами?
3. Какова структура процедур?
4. Какова структура подпрограммы-функции?

Тема 8 Динамические структуры и управление памятью в языке Паскаль**Контрольные вопросы:**

1. Файлы. Файловые типы.
2. Понятие прямого и последовательного методов доступа.
3. Файловая система Турбо Паскаля: понятие логического и физического файлов. Текстовые файлы.

Вопросы для контроля знаний:

1. Типизированные файлы.
2. Процедуры и функции для организации прямого метода доступа.
3. Алгоритмы формирования, просмотра и корректировки текстового и типизированного файлов.
4. Бестиповые файлы.

Лабораторная работа № 4 Разработка программ линейной, разветвляющейся, циклической структуры. Составление подпрограмм – процедур, подпрограмм – функций

Цель работы: Ознакомиться со структурой программы, научиться программировать простейшие структуры

Вопросы по лабораторной работе 4

1. В чем состоит различие и сходство подпрограмм-процедур и подпрограмм-функций?
2. Как обратиться к процедурам и функциям?
3. Что называется параметром и каково его назначение?
4. Назначение формальных и фактических параметров и их взаимосвязь.
5. Опишите последовательность событий при вызове процедур или функций.
6. Для чего используется пошаговый режим с заходом в процедуры и как его осуществить?
7. Каковы отличия параметров-значений от параметров-переменных, особенности их описания и применения?

8. Чем отличаются локальные и глобальные параметры? Какова область их действия?
9. Какой алгоритм является алгоритмом циклической структуры?
10. Типы циклов в языке.
11. Какой цикл выполнится хотя бы один раз? А какой - ни разу?

Тема 9 Принцип объектно-ориентированного подхода к разработке программного обеспечения

Контрольные вопросы:

1. Ссылочные типы: стандартный и пользовательский. Действия с указателями.
2. Динамические структуры данных. Схема распределения памяти при выполнении программ.
3. Действия с динамическими переменными: создание и уничтожение. Динамические массивы и массивы указателей.
4. Функции для определения свободной динамической памяти.

Вопросы для контроля знаний:

1. Линейные списки.
2. Формирование, просмотр, удаление и включение элемента.
3. Примеры программ.
4. Стек. Формирование, добавление, исключение элемента из стека.
5. Примеры программ.
6. Очереди. Включение элемента в очередь и удаление.
7. Двухнаправленные списки.
8. Создание списка, удаление и вставка элемента.

Лабораторная работа № 5 Реализация программ использующих основные структуры программирования

Цель работы: Ознакомиться с основными структурами программирования, научиться составлять программы с разветвляющимися и циклическими структурами

Вопросы по лабораторной работе 5

1. Какой алгоритм является алгоритмом разветвляющейся структуры?
2. Что такое условный оператор? Оператор выбора?
3. Полная и сокращенная формы условного оператора.
4. Что такое составной оператор? Формат его записи.
5. В каких случаях применяется полная форма условного оператора?
6. Что такое метка? Описывается ли метка в программе?
7. Что такое оператор безусловного перехода?
8. В каких случаях нельзя обойтись без операторов безусловного перехода?
9. В каких случаях надо использовать CASE вместо IF?

Тема 10 Конструирование, тестирование, и отладка программ

Контрольные вопросы:

1. Тестирование модулей и интегральное тестирование.
2. Формирование тестовых данных: классов, этапов тестирования, анализа результатов.
3. Тестовые мониторы. Виды отладки.
4. Языки и системы тестирования.
5. Отладчики и отладочная информация.

Вопросы для контроля знаний:

1. Методы разработки структуры программы.
2. Иерархические схемы.
3. Правила детализации.
4. Сегментирование.
5. Структурное и функциональное тестирование.

Лабораторная работа № 6 Разработка программ с переменными типа запись. Работа с файлами данных

Цель работы: Научиться составлять алгоритмы различных структур. Познакомиться с неоднородными структурами данных – записи, файлы.

Вопросы по лабораторной работе 6

1. Тип запись. Объявление.
2. Принципы обработки.
3. Определение файла.
4. Классификация файлов.
5. Процедуры и функции для обработки типизированных файлов.

Тема 11 Языки и системы программирования.

Контрольные вопросы:

1. Языки программирования высокого уровня.
2. Консольные приложения.
3. Чтение и вывод данных.

Вопросы для контроля знаний:

1. Ключевые слова.
2. Типы данных

Лабораторная работа № 7 Создание программ с переменными типа запись

Цель работы: Научиться составлять алгоритмы структуры типа запись

Вопросы по лабораторной работе 7

1. Как описываются переменные типа запись?
2. В каких случаях целесообразно использовать переменные типа запись?
3. Из каких компонентов состоит переменная типа запись?
4. Каково назначение оператора WITH?
5. Какие операции допустимы над полями записи?
6. В чем отличие записей от массивов?

Тема 12 Операторы и выражения.

Контрольные вопросы:

1. Условный оператор.
2. Оператор цикла.
3. Возможности вывода.
4. Литералы.
5. Управляющие последовательности.

Вопросы для контроля знаний:

1. Область видимости переменных.
2. Операторы.
3. Условные логические операторы.
4. Оператор присваивания.

5. Поразрядные операторы

Лабораторная работа № 8 Манипулирование адресами области памяти с помощью указателей

Цель работы: Изучить особенности работы с динамическими переменными, работы с оперативной памятью, познакомиться со стандартными процедурами и функциями для работы с динамическими переменными

Вопросы по лабораторной работе 8

1. Дайте понятие указателя. Для каких целей он служит?
2. Поясните следующие понятия: установка указателя, тип указателя.
3. Какие действия выполняют операции * и & ?
4. Какие действия необходимо выполнить, чтобы обратиться к переменной через указатель (начиная с объявления указателя).
5. Что такое адресная арифметика?
6. Какие ограничения действуют в адресной арифметике на операцию «присваивания»?
7. Какие ограничения действуют в адресной арифметике на операции «инкремента» и «декремента»?
8. Какие ограничения действуют в адресной арифметике на «сложение» и «вычитание»?
9. Что представляет собой массив данных с точки зрения указателей?
10. Перечислите три направления применения указателей в параметрах функций.
11. Как функция может обратиться к памяти, находящейся за её пределами? Приведите пример.

Тема 13 Управляющие операторы

Контрольные вопросы:

1. Управляющие операторы.
2. Оператор if.
3. Вложенный оператор if.
4. Оператор switch.

Вопросы для контроля знаний:

1. Оператор цикла for.
2. Оператор цикла while.
3. Оператор break.
4. Оператор continue.

Перечисления

Лабораторная работа № 9 Разработка и реализация программ с файлами данных

Цель работы: Научиться составлять алгоритмы структуры типа файл

Вопросы по лабораторной работе 9

1. Файлы. Файловые типы. Понятие прямого и последовательного методов доступа.
2. Файловая система Турбо Паскаля: понятие логического и физического файлов.
3. Текстовые файлы.
4. Типизированные файлы. Процедуры и функции для организации прямого метода доступа.
5. Алгоритмы формирования, просмотра и корректировки текстового и типизированного файлов.
6. Бестиповые файлы.

Тема 14 Массивы и строки. Введение в классы, объекты и методы. Файлы и потоки

Контрольные вопросы:

1. Одномерные массивы.
2. Инициализация массива.
3. Двумерный массив.
4. Многомерные массивы.
5. Ступенчатые массивы.
6. Определение класса.
7. Переменные ссылочного типа и присваивание.

Вопросы для контроля знаний:

1. Методы.
2. Возврат из метода.
3. Уровни доступа.
4. Использование параметров.
5. Исключение недоступного кода.
6. Конструкторы.
7. Параметризованные конструкторы.
8. Сборка мусора и деструкторы.
9. Классы потоков.
10. Консольный ввод-вывод.
11. Класс FileStream.
12. Запись в файл.
13. Символьный ввод-вывод в файл.
14. Применение класса StreamWriter.
15. Применение класса StreamReader

Лабораторная работа № 10 Динамическое программирование

Цель работы: Изучить особенности работы с динамическими переменными, работы с оперативной памятью, познакомиться со стандартными процедурами и функциями для работы с динамическими переменными

Вопросы по лабораторной работе 10

1. Для каких оптимизационных задач применяется метод динамического программирования?
2. В чем заключается суть метода динамического программирования?
3. Сформулируйте принцип оптимальности Беллмана.
4. Что является целевой функцией в задаче о кратчайшем маршруте?
5. Какой параметр определяет состояние системы на каждом шаге?

Тема 15 Виды классов Разработка Windows-приложений. ООП в деталях и примерах

Контрольные вопросы:

1. Событийное управление.
2. Свойства окна.
3. Элементы интерфейса.
4. События на форме.
5. Ошибки на форме.
6. События кнопки Button.
7. Поле ввода TextBox.

Вопросы для контроля знаний:

1. События TextVox.
2. Виды классов, индексы и свойства.
3. Статические члены класса.
4. Константы и неизменяемые поля.
5. Статические неизменяемые поля.
6. Вложенные классы.
7. Наследование.
8. Инициализаторы конструкторов.
9. Использование интерфейсов.
10. Изолированные классы.
11. Абстрактные классы.
12. Перегрузка.

Лабораторная работа № 11 Разработка рекурсивных алгоритмов

Цель работы: Изучить особенности работы с рекурсивными процедурами и функциями.

Вопросы по лабораторной работе 11

1. Что такое рекурсивная функция?
2. Что такое прямая рекурсия?
3. Что такое косвенная рекурсия?
4. На каком-то шаге должен быть прекращен дальнейший вызов рекурсивной функции?

Тема 16 Виды классов. ООП в деталях и примерах. Наследование. Разработка Windows-приложений**Контрольные вопросы:**

1. UML-схема.
2. Отношения.
3. Обобщение.
4. Ассоциации.
5. Простые зависимости.
6. Одиночное наследование.
7. Отношение наследования.
8. Структурные отношения.
9. Диаграммы классов.

Вопросы для контроля знаний:

1. Наследование.
2. Доступ к членам класса.
3. Организация защищенного доступа.
4. Конструкторы и наследование.
5. Вызов конструкторов базового класса.
6. Наследование и сокрытие имен.
7. Создание многоуровневой иерархии классов.
8. Порядок вызова конструкторов.
9. Ссылки на базовый класс и объекты производных классов.

Лабораторная работа № 12 Реализация рекурсивных структур. Применение рекурсивных алгоритмов для различных сортировок

Цель работы: Изучить особенности работы с рекурсивными процедурами и функциями

Вопросы по лабораторной работе 12

1. Можно ли случай косвенной рекурсии свести к прямой рекурсии?

2. Может ли рекурсивная база содержать несколько тривиальных случаев?
3. Являются ли параметры, база и декомпозиция единственными для конкретной задачи?
4. С какой целью в задачах происходит пересмотр или корректировка выбранных параметров, выделенной базы или случая декомпозиции?
5. Является ли рекурсия универсальным способом решения задач?
6. Почему для оценки трудоемкости рекурсивного алгоритма недостаточно одного метода подсчета вершин рекурсивного дерева?

Тема 17 Интерфейсы и структуры.

Контрольные вопросы:

1. Интерфейсы.
2. Интерфейсные индексы.
3. Интерфейсные свойства.

Вопросы для контроля знаний:

1. Явные реализации.
2. Стандартные интерфейсы для среды .NET Framework.
3. Структуры

Лабораторная работа № 13 Модульная структура программы. Отладка модулей

Цель работы: Познакомиться с модульным программированием, со стандартными модулями языка. Научиться программировать модули

Вопросы по лабораторной работе 13

1. Что называется программным средством?
2. Что понимается под надежностью программного средства?
3. Могут ли в надежном программном средстве быть ошибки?
4. Что понимается под технологией программирования?
5. Что такое система? Какие бывают системы?
6. В чем заключается природа ошибок человека при преобразовании информации из одной формы в другую и каковы пути борьбы с этими ошибками?
7. Что входит в разработку программного средства?
8. Что понимают под качеством программного средства?
9. Перечислите критерии качества программного средства.
10. Что понимается под функциональностью (надежностью, легкостью, эффективностью, сопровождаемостью, мобильностью) программного средства?
11. Назовите общие принципы обеспечения надежности программного средства.
12. Какие действия предпринимают для предупреждения ошибок при разработке программных средств?
13. Какие методы используются для борьбы со сложностью программных средств?
14. Как обеспечивается точность перевода документов различными разработчиками программного средства?
15. Что нужно для преодоления барьера между пользователем и разработчиком?
16. Что называют функциональной спецификацией, а что спецификацией качества программного средства?
17. Что такое примитив качества программного средства?
18. Перечислите примитивы качества программного средства. С какими критериями качества они связаны?
19. Дайте определения используемым примитивам качества программных средств.
20. Из чего состоит функциональная спецификация?

21. Что такое модульное программирование (программный модуль)?
22. Какие критерии используются для оценки программного модуля?
23. Чем измеряется размер программного модуля?
24. Что такое прочность программного модуля? Приведите примеры различных по степени прочности модулей.
25. Что такое сцепление модуля?
26. Какой вид сцепления программных модулей рекомендуется современной технологией программирования?
27. Какой модуль называется рутинным (зависящим от предыстории)?
28. В чем состоит метод восходящей (нисходящей) разработки?
29. Какие методы используются для контроля структуры программы? В чем их суть?
30. Назовите порядок разработки программного модуля. Что делается на каждом шаге разработки?
31. Какое программирование называют структурным?
32. Для чего используется язык псевдокода?
33. Какое программирование называется защитным? В чем оно заключается?
34. Перечислите рекомендации по обеспечению ясности текста (эффективности) модуля.
35. Какие методы применяют для контроля программного модуля?
36. Что понимается под отладкой (тестированием) программного средства?
37. Какие виды отладки программных средств приняты в нашей стране?
38. Какие рекомендации по отладке программного средства Вы знаете?
39. Как осуществляется автономное тестирование (тестирование при комплексной отладке)?

Тема 18 Делегаты, события и лямбда-выражения

Контрольные вопросы:

1. Делегаты.
2. Групповая адресация.
3. Групповое преобразование делегируемых методов.
4. Ковариантность и контравариантность.
5. Анонимные методы.
6. Передача аргументов анонимному методу.
7. Возврат значения из анонимного метода.
8. Лямбда-выражения.
9. Одиночные лямбда-выражения.

Вопросы для контроля знаний:

1. Блочные лямбда-выражения.
2. Коллекции, перечислители, итераторы Коллекции.
3. Типы коллекций.
4. Перечислители.
5. Итераторы.
6. Необобщенные коллекции.
7. Интерфейсы необобщенных коллекций.
8. Интерфейс ICollection.
9. Интерфейс IList.
10. Интерфейс IDictionary

Лабораторная работа № 14 Разработка и реализация модульных программ

Цель работы: Познакомиться с модульным программированием, со стандартными модулями Pascal

Вопросы по лабораторной работе 14

1. Делегаты.
2. Групповая адресация.
3. Групповое преобразование делегируемых методов.
4. Ковариантность и контравариантность.
5. Анонимные методы.
6. Передача аргументов анонимному методу.
7. Возврат значения из анонимного метода.
8. Лямбда-выражения.
9. Одиночные лямбда-выражения.
10. Блочные лямбда-выражения.

Задания для самостоятельной работы:

1. На плоскости заданы n точек своими координатами $(x_1, y_1), (x_2, y_2)$. Составить программу вычисления максимального внутреннего и минимального внешнего радиусов кольца с центром в начале координат, содержащего все точки. Координаты точек считывать из файла.
2. Составить программу, которая вводит натуральное число N и выдает все трехзначные числа, сумма цифр которых равна $N*3$.
3. Составить программу, которая считывает с командной строки имя файла и выводит на экран количество символов, строк и слов в этом файле.
4. Программа должна считывать с командной строки имена двух файлов и выводить на экран те строки этих файлов, которые отличаются друг от друга.
5. Программа, принимающая имя файла и выводящая его строки, перед каждой из которых записывается ее номер.
6. Программа считывает из файла матрицу и выводит на экран суммы столбцов этой матрицы.
7. Программа считывает с командной строки имя файла и слово и проверяет, встречается ли такое слово в файле.
8. Программа принимает имя файла, сортирует его строки и выдает их на экран.
9. Программа считывает с командной строки имя файла и выводит на экран все слова, встречающиеся в этом файле, без повторов.
10. Программа считывает с командной строки имя файла и выводит на экран наиболее часто встречающееся в нем слово.
11. Составить программу, которая запрашивает у пользователя двузначное целое число, вводит его и отображает на экране величину числа словами. Например, введено -12. Результат: минус двенадцать.
12. Программа принимает имя файла и символ и выводит на экран те строки файла, которые начинаются с указанного символа.
13. Программа для преобразования единиц измерения предназначена для преобразования физических величин друг в друга (метры в километры или сантиметры, килограммы в тонны, граммы и фунты). Программа принимает в командной строке число, исходную единицу измерения и целевую единицу измерения и выводит на экран преобразованное число. Единицы измерения задавать строками ("m" - для метров, "km" - километры, "cm" - сантиметры, "kg" - килограммы, "t" - тонны, "g" - граммы, "p" - фунты)
14. Программа, которая по заданному числу N печатает список всех простых чисел, не превышающих N .
15. Программа, которая по заданному числу N печатает список всех совершенных чисел, не превышающих N . Совершенным называется число, которое равно сумме своих делителей, например $6 = 1+2+3$ или $24 = 1 + 2 + 3 + 4 + 6 + 8$.

16. Программа считывает с командной строки имя файла и два слова и распечатывает на экране содержимое файла, в котором все вхождения первого слова заменены на второе.

17. Программа, которая выводит на экран все алфавитно-цифровые последовательности, содержащиеся в указанном файле. Алфавитно-цифровой называется непрерывная последовательность цифр и букв латинского алфавита. Пример: если файл содержит символы `asdf-+^&@qwerty!@#` то результатом работы программы должно быть: `asdfqwerty`

18. Программа принимает в командной строке имя файла и один из двух символов 'u' или 'l'. В зависимости от того, какой из символов передан, она преобразует содержимое файла к верхнему или нижнему регистру и распечатывает на экране.

19. Программа считывает из файла матрицу и выводит на экран след этой матрицы (сумму диагональных элементов).

20. Программа считывает две матрицы из файлов и записывает в третий файл матрицу, являющуюся их суммой.

21. Программа считывает из одного файла матрицу, а из другого – вектор и записывает в третий файл результат умножения матрицы на вектор.

22. Файл содержит записи следующего формата: операция число, где - одно из двух слов "inc" или "dec", а представляет собой некоторое целое число. Программа должна по заданному файлу с такими записями проверить, что сумма чисел, соответствующих слову "inc" равна сумме чисел для "dec".

Итоговый тест по дисциплине «Технология программирования»

1. Какие программы можно отнести к системному программному обеспечению:

1. операционные системы;
2. прикладные программы;
3. игровые программы.

2. Какие программы можно отнести к системному ПО:

1. драйверы;
2. текстовые редакторы;
3. электронные таблицы;
4. графические редакторы.

3. Специфические особенности ПО как продукта:

1. продажа по ценам ниже себестоимости (лицензирование);
2. низкие материальные затраты при создании программ;
3. возможность создание программ небольшие коллективом или даже одним человеком;
4. разнообразие решаемых задач с помощью программных средств.

4. Какие программы можно отнести к системному ПО:

1. программа расчета заработной платы;
2. электронные таблицы;
3. СУБД (системы управления базами данных).

5. Какие программы нельзя отнести к системному ПО:

1. игровые программы;
2. компиляторы языков программирования;
3. операционные системы;
4. системы управления базами данных.

6. Какие программы можно отнести к прикладному программному обеспечению:

1. электронные таблицы;
2. таблицы решений;
3. СУБД (системы управления базами данных).

7. Какие программы можно отнести к прикладному ПО:

1. программа расчета заработной платы;
2. диспетчер программ;

3. программа «Проводник» (Explorer).
8. Какие программы нельзя отнести к прикладному ПО:
 1. компиляторы и (или) интерпретаторы;
 2. текстовые и (или) графические редакторы;
 3. электронные таблицы.
9. Можно ли отнести операционную систему к программному обеспечению:
 1. да;
 2. нет.
10. Можно ли отнести операционную систему к прикладному программному обеспечению:
 1. да;
 2. нет.
11. Специфические особенности ПО как продукта:
 1. низкие затраты при дублировании;
 2. универсальность;
 3. простота эксплуатации;
 4. наличие поддержки (сопровождения) со стороны разработчика.
12. Какие программы можно отнести к системному ПО:
 1. утилиты;
 2. экономические программы;
 3. статистические программы;
 4. мультимедийные программы.
13. Этап, занимающий наибольшее время, в жизненном цикле программы:
 1. сопровождение;
 2. проектирование;
 3. тестирование;
 4. программирование;
 5. формулировка требований.
14. Этап, занимающий наибольшее время, при разработке программы:
 1. тестирование;
 2. сопровождение;
 3. проектирование;
 4. программирование;
 5. формулировка требований.
15. Первый этап в жизненном цикле программы:
 1. формулирование требований;
 2. анализ требований;
 3. проектирование;
 4. автономное тестирование;
 5. комплексное тестирование.
16. Один из необязательных этапов жизненного цикла программы:
 1. оптимизация;
 2. проектирование;
 3. тестирование;
 4. программирование;
 5. анализ требований.
17. Самый большой этап в жизненном цикле программы:
 1. эксплуатация;
 2. изучение предметной области;
 3. программирование;
 4. тестирование;
 5. корректировка ошибок.

18. Какой этап выполняется раньше:
 1. отладка;
 2. тестирование.
19. Какой этап выполняется раньше:
 1. отладка;
 2. оптимизация;
 3. программирование;
 4. тестирование.
20. Что выполняется раньше:
 1. компиляция;
 2. отладка;
 3. компоновка;
 4. тестирование.
21. Что выполняется раньше:
 1. проектирование;
 2. программирование;
 3. отладка;
 4. тестирование.
22. В стадии разработки программы не входит:
 1. автоматизация программирования;
 2. постановка задачи;
 3. составление спецификаций;
 4. эскизный проект;
 5. тестирование.
23. Самый важный критерий качества программы:
 1. работоспособность;
 2. надежность;
 3. эффективность;
 4. быстродействие;
 5. простота эксплуатации.
24. Способы оценки качества:
 1. сравнение с аналогами;
 2. наличие документации;
 3. оптимизация программы;
 4. структурирование алгоритма.
25. Существует ли связь между эффективностью и оптимизацией программы:
 1. да;
 2. нет.
26. Наиболее важный критерий качества:
 1. надежность;
 2. быстродействие;
 3. удобство в эксплуатации;
 4. удобный интерфейс;
 5. эффективность.
27. Способы оценки надежности:
 1. тестирование;
 2. сравнение с аналогами;
 3. трассировка;
 4. оптимизация.
28. Повышает ли качество программ оптимизация:
 1. да;

2. нет.
29. Существует ли связь между надежностью и быстродействием:
 1. нет;
 2. да.
30. В каких единицах можно измерить надежность:
 1. отказов/час;
 2. км/час;
 3. Кбайт/сек;
 4. операций/сек.
31. В каких единицах можно измерить быстродействие:
 1. отказов/час;
 2. км/час;
 3. Кбайт/сек;
 4. операций/сек.
32. Что относится к этапу программирования:
 - А)написание кода программы;
 - В) разработка интерфейса;
 - С) работоспособность;
 - Д)анализ требований.
33. Последовательность этапов программирования:
 - А)компилирование, компоновка, отладка;
 - В) компоновка, отладка, компилирование;
 - С)отладка, компилирование, компоновка;
 - Д)компилирование, отладка, компоновка.
- 34) Инструментальные средства программирования:
 1. компиляторы, интерпретаторы;
 2. СУБД (системы управления базами данных);
 3. BIOS (базовая система ввода-вывода);
 4. ОС (операционные системы).
35. На языке программирования составляется:
 1. исходный код;
 2. исполняемый код;
 3. объектный код;
 4. алгоритм.
36. Правила, которым должна следовать программа это:
 1. алгоритм;
 2. структура;
 3. спецификация;
 4. состав информации.
37. Можно ли внутри цикла поместить еще один цикл:
 1. да;
 2. нет.
38. Можно ли внутри условного оператора поместить еще одно условие:
 1. да;
 2. нет.
39. Можно ли одно большое (длинное) выражение разбить на несколько выражений:
 1. да;
 2. нет.
40. Если имеется стандартная функция, нужно ли писать собственную:
 1. нет;
 2. да.

41. Доступ, при котором записи файла читаются в физической последовательности, называется:
1. последовательным;
 2. прямым;
 3. простым;
 4. основным.
42. Доступ, при котором записи файла обрабатываются в произвольной последовательности, называется:
1. прямым;
 2. последовательным;
 3. простым;
 4. основным.
43. Методы программирования (укажите НЕ верный ответ):
1. логическое;
 2. структурное;
 3. модульное.
44. Что выполняется раньше:
1. разработка алгоритма;
 2. выбор языка программирования;
 3. написание исходного кода;
 4. компиляция.
45. Можно ли переменным присваивать произвольные идентификаторы:
1. да;
 2. нет.
46. Найдите НЕ правильное условие для создания имен:
1. имена могут содержать пробелы;
 2. длинное имя можно сократить;
 3. из имени лучше выбрасывать гласные;
 4. можно использовать большие буквы.
47. Какие символы не допускаются в именах переменных:
1. пробелы;
 2. цифры;
 3. подчеркивание
48. Можно ли использовать имена, которые уже были использованы в другой программе (модуле):
1. да;
 2. нет.
49. Можно ли ставить знак подчеркивания в начале имени:
1. да, но не рекомендуется;
 2. да, без ограничений;
 3. нет.
50. Как называется способ составления имен переменных, когда в начале имени сообщается тип переменной:
5. прямым указанием;
 6. венгерской нотацией;
 7. структурным программированием;
 8. поляризацией.
51. Можно ли писать комментарии в отдельной строке:
1. да;
 2. нет.
52. Транслируются ли комментарии:
1. да;

2. нет.
53. Наличие комментариев позволяет:
 1. быстрее найти ошибки в программе;
 2. быстрее писать программы;
 3. быстрее выполнять программы.
54. Наличие комментариев позволяет:
 1. легче разобраться в программе;
 2. применять сложные структуры;
 3. увеличить быстродействие.
55. Наличие комментариев позволяет:
 1. улучшить читабельность программы;
 2. улучшить эксплуатацию программы;
 3. повысить надежность программы.
56. Что определяет выбор языка программирования:
 1. область приложения;
 2. знание языка;
 3. наличие дополнительных библиотек.
57. Возможно ли комбинирование языков программирования в рамках одной задачи:
 1. да;
 2. нет.
58. Допустимо ли комбинирование языков программирования в рамках одной задачи :
 1. да;
 2. нет.
59. Для каких задач характерно использование большого количества исходных данных, выполнение операций поиска, группировки:
 1. для экономических задач;
 2. для системных задач;
 3. для инженерных задач.

60. Для каких задач характерен большой объем вычислений, использование сложного математического аппарата:
 1. для инженерных задач;
 2. для системных задач;
 3. для экономических задач.
61. На каком этапе производится выбор языка программирования:
 1. проектирование;
 2. программирование;
 3. отладка;
 4. тестирование.
62. Можно ли использовать комбинацию языков программирования в рамках одного проекта:
 1. да;
 2. нет.
63. Для решения экономических задач характерно применение:
 1. СУБД (систем управления базами данных);
 2. языков высокого уровня;
 3. языков низкого уровня;
 4. применение сложных математических расчетов.
64. Для решения инженерных задач характерно применение:
 1. САПР (систем автоматизированного проектирования);
 2. СУБД (систем управления базами данных);
 3. ОС (операционных систем).

65. Причины синтаксических ошибок:
1. плохое знание языка программирования;
 2. ошибки в исходных данных;
 3. ошибки, допущенные на более ранних этапах;
 4. неправильное применение процедуры тестирования.
66. Когда можно обнаружить синтаксические ошибки:
1. при компиляции;
 2. при отладке;
 3. при тестировании;
 4. на этапе проектирования;
 5. при эксплуатации.
67. Ошибки компоновки заключаются в том, что:
1. указано внешнее имя, но не объявлено;
 2. неправильно использовано зарезервированное слово;
 3. составлено неверное выражение;
 4. указан неверный тип переменной.
68. Могут ли проявиться ошибки при изменении условий эксплуатации:
1. да;
 2. нет.
69. Могут ли проявиться ошибки при изменении в предметной области:
1. да;
 2. нет.
70. Возможно ли программирование с защитой от ошибок:
1. да;
 2. нет.
71. Есть ли недостатки программирования с защитой от ошибок:
1. да;
 2. нет.
72. Защитное программирование это:
1. встраивание в программу отладочных средств;
 2. создание задач защищенных от копирования;
 3. разделение доступа в программе;
 4. использование паролей;
 5. оформление авторских прав на программу.
73. Вид ошибки с неправильным написанием служебных слов (операторов):
1. синтаксическая;
 2. семантическая;
 3. логическая;
 4. символьная.
74. Вид ошибки с неправильным использованием служебных слов (операторов):
1. семантическая;
 2. синтаксическая;
 3. логическая;
 4. символьная.
75. Ошибки при написании программы бывают:
1. синтаксические;
 2. орфографические;
 3. лексические;
 4. фонетические;
 5. морфологические.
76. Процедура поиска ошибки, когда известно, что она есть это:
1. отладка;

2. тестирование;
3. компоновка;
4. транзакция;
5. трансляция.
6. 77. Программа для просмотра значений переменных при выполнении программы:
7. отладчик;
8. компилятор;
9. интерпретатор;
10. трассировка;
11. тестирование.
78. Отладка – это:
 1. процедура поиска ошибок, когда известно, что ошибка есть;
 2. определение списка параметров;
 3. правило вызова процедур (функций);
 4. составление блок-схемы алгоритма.
79. Когда программист может проследить последовательность выполнения команд программы:
 1. при трассировке;
 2. при тестировании;
 3. при компиляции;
 4. при выполнении программы;
 5. при компоновке.
80. На каком этапе создания программы могут появиться синтаксические ошибки:
 1. программирование;
 2. проектирование;
 3. анализ требований;
 4. тестирование.
81. Когда приступают к тестированию программы:
 1. когда программа уже закончена;
 2. после постановки задачи;
 3. на этапе программирования;
 4. на этапе проектирования;
 5. после составления спецификаций,
82. Тестирование бывает:
 1. автономное;
 2. инструментальное;
 3. визуальное;
 4. алгоритмическое.
83. Тестирование бывает:
 1. комплексное;
 2. инструментальное;
 3. визуальное;
 4. алгоритмическое.
84. Существует ли различие между отладкой и тестированием:
 1. да;
 2. нет.
85. При комплексном тестировании проверяются:
 1. согласованность работы отдельных частей программы;
 2. правильность работы отдельных частей программы;
 3. быстродействие программы;
 4. эффективность программы.
86. Чему нужно уделять больше времени, чтобы получить хорошую программу:

1. тестированию;
 2. программированию;
 3. отладке;
 4. проектированию.
87. Процесс исполнения программы с целью обнаружения ошибок:
1. тестирование;
 2. кодирование;
 3. сопровождение;
 4. проектирование.
88. Автономное тестирование это:
1. тестирование отдельных частей программы;
 2. инструментальное средство отладки;
 3. составление блок-схем;
 4. пошаговая проверка выполнения программы.
89. Трассировка это:
1. проверка пошагового выполнения программы;
 2. тестирование исходного кода;
 3. отладка модуля;
 4. составление блок-схемы алгоритма.
90. Локализация ошибки:
1. определение места возникновения ошибки;
 2. определение причин ошибки;
 3. обнаружение причин ошибки;
 4. исправление ошибки.
91. Назначение тестирования:
1. повышение надежности программы;
 2. обнаружение ошибок;
 3. повышение эффективности программы;
 4. улучшение эксплуатационных характеристик;
 5. приведение программы к структурированному виду.
92. Назначение отладки:
1. поиск причин существующих ошибок;
 2. поиск возможных ошибок;
 3. составление спецификаций;
 4. разработка алгоритма.
93. Инструментальные средства отладки (НЕ правильный ответ):
1. компиляторы;
 2. отладчики;
 3. трассировка.
94. Отладка программ это:
1. локализация и исправление ошибок;
 2. алгоритмизация программирования;
 3. компиляция и компоновка.
95. Что выполняется раньше, автономная или комплексная отладка:
1. автономная;
 2. комплексная.
96. Что выполняется раньше, отладка или тестирование:
1. отладка;
 2. тестирование.
97. Что такое автоматизация программирования:
1. создание исходного кода программными средствами;
 2. создание исходного кода при помощи компилятора;

3. создание исходного кода без разработки алгоритма.
98. В чем сущность автоматизации программирования:
1. создание программы без написания ее текста;
 2. получение готовой программы без выполнения компоновки;
 3. в отсутствии компиляции.
99. Возможна ли автоматизация программирования:
1. да;
 2. нет.
100. Создание исполняемого кода программы без написания исходного кода называется:
1. составлением спецификаций;
 2. отладкой;
 3. проектированием.
 4. автоматизацией программирования;
101. Одно из преимуществ автоматизации программирования:
1. наглядное программирование с визуальным контролем;
 2. получение стандартной программы;
 3. создание программы с оптимальным кодом.
102. Один из методов автоматизации программирования:
1. структурное программирование;
 2. модульное программирование;
 3. визуальное программирование;
 4. объектно-ориентированное программирование.
103. Влияет ли автоматизация программирования на эффективность программы:
1. нет;
 2. да
104. Автоматизация программирования позволяет:
1. повысить надежность программы;
 2. сократить время разработки программы;
 3. повысить быстродействие программы.
105. Позволяет ли автоматизация программирования всегда создавать эффективные программы:
1. да.
 2. нет;
106. Позволяет ли автоматизация программирования всегда создавать надежные программы:
- A) нет;
 - B) да.
107. Недостаток автоматизации программирования;
- A) большой размер программы;
 - B) низкое быстродействие;
 - C) сложность программы.
108. Возможны ли ошибки при автоматизации программирования:
- A) да;
 - B) нет.
109. Нужно ли выполнять тестирование при автоматизации программирования:
- A) да;
 - B) нет.
110. Выполняется ли процедура компиляции при автоматизации программирования:
- A) да;
 - B) нет.
111. Что легко поддается автоматизации:
- A) интерфейс;

- В) работа с файлами;
 - С) сложные логические задачи;
 - Д) алгоритмизация.
112. Относится ли визуальное программирование к средствам автоматизации:
- А) да;
 - В) нет.
113. Нахождение наилучшего варианта из множества возможных:
- А) оптимизация;
 - В) тестирование;
 - С) автоматизация;
 - Д) отладка;
 - Е) сопровождение.
114. Что такое оптимизация программ:
- А) улучшение работы существующей программы;
 - В) создание удобного интерфейса пользователя;
 - С) разработка модульной конструкции программы;
 - Д) применение методов объектно-ориентированного программирования.
115. Критерии оптимизации:
- А) время выполнения или размер требуемой памяти;
 - В) размер программы и ее эффективность;
 - С) независимость модулей;
 - Д) качество программы, ее надежность.
116. Критерии оптимизации:
- А) эффективность использования ресурсов;
 - В) структурирование алгоритма;
 - С) структурирование программы.
117. Возможна ли оптимизация программ без участия программиста:
- А) да;
 - В) нет.
118. Возможна ли оптимизация циклов:
- А) да;
 - В) нет.
119. В чем заключается оптимизация условных выражений:
- А) в изменении порядка следования элементов выражения;
 - В) в использовании простых логических выражений;
 - С) в использовании сложных логических выражений;
 - Д) в использовании операций AND, OR и NOT.
120. Оптимизация циклов заключается в:
- А) уменьшении количества повторений тела цикла;
 - В) просмотре задачи с другой стороны;
 - С) упрощение задачи за счет включения логических операций.
121. Оптимизация программы это:
- А) модификация;
 - В) отладка;
 - С) повышение сложности программы;
 - Д) уменьшение сложности программы.
122. Критерии оптимизации программы:
- А) быстродействие или размер программы;
 - В) быстродействие и размер программы;
 - С) надежность или эффективность;
 - Д) надежность и эффективность.
123. Результат оптимизации программы:

- A) эффективность;
 - B) надежность;
 - C) машино-независимость;
 - D) мобильность.
124. Сущность оптимизации циклов:
- A) сокращение количества повторений выполнения тела цикла;
 - B) сокращение тела цикла;
 - C) представление циклов в виде блок-схем;
 - D) трассировка циклов;
 - E) поиск ошибок в циклах.
125. В чем сущность модульного программирования:
- A) в разбиении программы на отдельные функционально независимые части;
 - B) в разбиении программы на отдельные равные части;
 - C) в разбиении программы на процедуры и функции;
126. Можно ли сочетать модульное и структурное программирование:
- A) да;
 - B) нет.
127. Может ли модуль включать несколько процедур или функций:
- A) да;
 - B) нет.
128. Рекомендуемые размеры модулей:
- A) небольшие;
 - B) большие;
 - C) равные;
 - D) фиксированной длины.
129. В чем заключается независимость модуля:
- A) в написании, отладке и тестировании независимо от остальных модулей;
 - B) в разработке и написании независимо от других модулей;
 - C) в независимости от работы основной программы.
130. При модульном программировании желательно, чтобы модуль имел:
- A) большой размер;
 - B) небольшой размер;
 - C) фиксированный размер;
 - D) любой размер.
131. Модульное программирование это:
- A) разбиение программы на отдельные части;
 - B) структурирование;
 - C) использование стандартных процедур и функций.
132. Можно ли использовать оператор GO TO в модульных программах:
- A) можно;
 - B) нельзя.
133. Разрешается ли использование циклов при модульном программировании:
- A) да;
 - B) нет.
134. Разрешается ли использование условных операторов при модульном программировании:
- A) да;
 - B) нет.
135. Сократится ли размер программы, если ее написать в виде набора модулей:
- A) нет;
 - B) да.
136. Достоинство модульного программирования:

- A) создание программы по частям в произвольном порядке;
 - B) не требует компоновки;
 - C) всегда дает эффективные программы;
 - D) снижает количество ошибок.
137. Недостаток модульного программирования:
- A) увеличивает трудоемкость программирования;
 - B) усложняет процедуру комплексного тестирования;
 - C) снижает быстродействие программы;
 - D) не позволяет выполнять оптимизацию программы.
138. Достоинство модульного программирования:
- A) возможность приступить к тестированию до завершения написания всей программы;
 - B) не требует комплексного тестирования;
 - C) уменьшает размер программы;
 - D) повышает надежность программы.
139. Допустимо ли использование оператора GO TO при структурном программировании:
- A) нет;
 - B) да.
140. Можно ли сочетать структурное программирование с модульным:
- A) можно;
 - B) нельзя;
 - C) только в особых случаях.
141. Любую ли программу можно привести к структурированному виду:
- A) любую;
 - B) не все;
 - C) нельзя.
142. Можно ли использовать оператор GO TO в структурированных программах:
- A) можно;
 - B) нельзя;
 - C) только в особых случаях.
143. Возможно, ли преобразовать неструктурированную программу к структурному виду:
- A) да;
 - B) нет.
144. Возможно ли программирование без оператора GO TO:
- A) да;
 - B) нет.
145. При структурном программировании задача выполняется:
- A) поэтапным разбиением на более легкие задачи;
 - B) без участия программиста;
 - C) объединением отдельных модулей программы.
146. Разрешается ли использование оператора GO TO при структурном программировании:
- A) нет;
 - B) да;
 - C) иногда.
147. Разрешается ли использование циклов при структурном программировании:
- A) да;
 - B) нет.
148. Разрешается ли использование оператора IF при структурном программировании:
- A) да;
 - B) нет.
149. Программирование без GO TO применяется. при:
- A) структурном программировании;
 - B) модульном программировании;

- С) объектно-ориентированном программировании;
 D) все ответы верные.
150. Достоинство структурного программирования:
 A) можно приступить к комплексному тестированию на раннем этапе разработки;
 B) можно приступить к автономному тестированию на раннем этапе разработки;
 C) нет необходимости выполнять тестирование;
 D) можно пренебречь отладкой.

2.2 Критерии оценки качества освоения дисциплины

Качество освоения дисциплины оценивается по степени успешности выполнения практических заданий, лабораторных практикумов и результатов прохождения тестирования.

Критерии оценки тестовых заданий, выполняемых студентами:

«Отлично»	Выполнение более 90% тестовых заданий
«Хорошо»	Выполнение от 65% до 90% тестовых заданий
«Удовлетворительно»	Выполнение более 50% тестовых заданий
«Неудовлетворительно»	Выполнение менее 50% тестовых заданий

Критерии оценки знаний обучающихся при выполнении лабораторных практикумов:

Оценка «5» ставится в том случае, если:

- лабораторная работа подготовлена к выполнению, обучаемый знает цель лабораторной работы;
- задания решены без ошибок с первого раза, правильно выбраны решения заданий;
- правильно выполнены расчёты, обучающийся понимает, что они значат;
- полно даны ответы на письменные и устные контрольные вопросы;
- отчёт оформлен аккуратно, сделаны выводы.

Оценка «4» ставится в том случае, если

- лабораторная работа подготовлена к выполнению, обучаемый знает цель лабораторной работы;
- задания решены с ошибками, потребовалась дополнительная помощь преподавателя, правильно выбраны методики решения заданий;
- расчёты выполнены с консультацией преподавателя;
- полно даны ответы на письменные и устные контрольные вопросы;
- отчёт оформлен аккуратно, сделаны выводы.

Оценка «3» ставится в том случае, если

- лабораторная работа подготовлена к выполнению, обучаемый знает цель лабораторной работы;
- задания выполнены с ошибками, потребовалась дополнительная помощь преподавателя, правильно выбраны методики решения заданий;
- с ошибками выполнены расчёты, даже с консультацией преподавателя или обучающийся не может объяснить, как выполнялись расчеты;
- даны ответы на письменные и устные контрольные вопросы.
- отчёт оформлен небрежно, сделаны выводы.

Оценка «2» ставится в том случае, если

- лабораторная работа подготовлена к выполнению, обучаемый не знает цель лабораторной работы;
- задачи решены с ошибками, потребовалась дополнительная помощь преподавателя, неверно выбраны методы решения задач;

- не выполнены расчёты;
- не даны ответы на устные контрольные вопросы;
- отчёт оформлен небрежно, выводы не сделаны.

Критерии оценки знаний обучающихся при выполнении практических заданий:

Оценка «отлично» – ставится, если студент демонстрирует знание теоретического и практического материала по теме практической работы, определяет взаимосвязи между показателями задачи, даёт правильный алгоритм решения, определяет междисциплинарные связи по условию задания. А также, если студент имеет глубокие знания учебного материала по теме практической работы, показывает усвоение взаимосвязи основных понятий используемых в работе, смог ответить на все уточняющие и дополнительные вопросы.

Оценка «хорошо» – ставится, если студент демонстрирует знание теоретического и практического материала по теме практической работы, допуская незначительные неточности при решении задач, имея неполное понимание междисциплинарных связей при правильном выборе алгоритма решения задания. А также, если студент показал знание учебного материала, усвоил основную литературу, смог ответить почти полно на все заданные дополнительные и уточняющие вопросы.

Оценка «удовлетворительно» – ставится, если студент затрудняется с правильной оценкой предложенной задачи, дает неполный ответ, требующий наводящих вопросов преподавателя, выбор алгоритма решения задачи возможен при наводящих вопросах преподавателя. А также, если студент в целом освоил материал практической работы, ответил не на все уточняющие и дополнительные вопросы.

Оценка «неудовлетворительно» – ставится, если студент дает неверную оценку ситуации, неправильно выбирает алгоритм действий. А также, если он имеет существенные пробелы в знаниях основного учебного материала практической работы, который полностью не раскрыл содержание вопросов, не смог ответить на уточняющие и дополнительные вопросы.

Критерии оценивания курсовой работы:

Оценка «отлично» выставляется при условии, что:

- работа выполнена самостоятельно, носит творческий характер, возможно содержание элементов научной новизны;
- собран, обобщен и проанализирован достаточный объем литературных источников;
- при написании и защите работы студентом продемонстрирован высокий уровень развития компетенций, теоретические знания и наличие практических навыков;
- работа хорошо оформлена и своевременно представлена на кафедру, полностью соответствует требованиям, предъявляемым к содержанию и оформлению курсовых работ;
- на защите освещены все вопросы исследования, ответы студента на вопросы профессионально грамотны, исчерпывающие.

Оценка «хорошо» ставится, если:

- тема работы раскрыта, однако выводы и рекомендации не всегда оригинальны и / или не имеют практической значимости, есть неточности при освещении отдельных вопросов темы;
- собран, обобщен и проанализирован необходимый объем психологической литературы, но не по всем аспектам исследуемой темы сделаны выводы и обоснованы практические рекомендации;
- при написании и защите работы студентом продемонстрирован средний уровень развития компетенций, наличие теоретических знаний и достаточных практических навыков;

- работа своевременно представлена на кафедру, есть отдельные недостатки в ее оформлении;
- в процессе защиты работы были неполные ответы на вопросы.

Оценка «удовлетворительно» ставится, когда:

- тема работы раскрыта частично, но в основном правильно, допущено поверхностное изложение отдельных вопросов темы;
- в работе недостаточно полно была использована техническая литература, выводы и практические рекомендации не отражали в достаточной степени содержание работы;
- при написании и защите работы студентом продемонстрирован удовлетворительный уровень развития компетенций, поверхностный уровень теоретических знаний и практических навыков;
- работа своевременно представлена на кафедру, однако не в полном объеме по содержанию и / или оформлению соответствует предъявляемым требованиям;
- в процессе защиты выпускник недостаточно полно изложил основные положения работы, испытывал затруднения при ответах на вопросы.

Оценка «неудовлетворительно» ставится, если:

- содержание работы не раскрывает тему, вопросы изложены бессистемно и поверхностно, нет анализа практического материала, основные положения и рекомендации не имеют обоснования;
- при написании и защите работы студентом продемонстрирован неудовлетворительный уровень развития компетенций;
- работа несвоевременно представлена на кафедру, не в полном объеме по содержанию и оформлению соответствует предъявляемым требованиям;
- на защите студент показал поверхностные знания по исследуемой теме, отсутствие представлений об актуальных проблемах по теме работы, плохо отвечал на вопросы.

3. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

3.1 Теоретические вопросы и практические задания для проведения экзамена

Вопросы для подготовки к экзамену

1. Основные конструкции алгоритмического языка
2. Жизненный цикл программы
3. Особенности языка программирования
4. Структура программы в языке
5. Понятие переменной
6. Объявление переменных.
7. Типы данных, определяемые пользователем.
8. Инициализация переменных.
9. Основные типы переменных
10. Синтаксис объявления имени переменной
11. Понятие константы. Объявление.
12. Понятие константы. Инициализация.
13. Простейшие арифметические операции
14. Операция %. Ее особенности.
15. Основные математические функции
16. Принципы структурного программирования
17. Оператор множественного ветвления switch
18. Условный оператор if-else
19. Вложенный оператор if.

20. Классификация циклов
21. Операции инкремента и декремента.
22. Оператор while()
23. Оператор For.
24. Оператор do...while()
25. Принципы модульного программирования
26. Описание функции
27. Вызов функции.
28. Прототипы функций.
29. Передача параметров по имени.
30. Передача параметров по адресу.
31. Данные типа «указатель».
32. Операции для работы с указателями
33. Объявление и инициализация одномерного массива.
34. Объявление и инициализация двумерного массива.
35. Доступ к элементам одномерного массива по индексу.
36. Доступ к элементам одномерного массива по адресу.
37. Доступ к элементам двумерного массива по индексу.
38. Доступ к элементам двумерного массива по адресу.
39. Принципы объектно-ориентированного программирования. Инкапсуляция.
40. Принципы объектно-ориентированного программирования. Полиморфизм.
41. Принципы объектно-ориентированного программирования. Наследование.
42. Структуры. Объявление.
43. Структуры. Инициализация.
44. Прямой доступ к элементам структуры.
45. Косвенный доступ к элементам структуры.
46. Массивы структур. Объявление.
47. Массивы структур. Инициализация.
48. Классы. Объявление открытых переменных.
49. Классы. Инициализация открытых переменных.
50. Классы. Объявление закрытых переменных.
51. Классы. Инициализация закрытых переменных.
52. Спецификаторы доступа.
53. Функции-члены класса.
54. Связывание функций.
55. Встроенные функции.
56. Конструкторы . Основные понятия.
57. Конструкторы по умолчанию.
58. Конструкторы с параметрами.
59. Стандартный класс string. Основные понятия.
60. Основные операции класса string.

Тематика курсовых работ

Вопросы теоретической части:

1. Теоретические проблемы разработки языков программирования. История языков программирования. Эволюция архитектуры программного обеспечения.
2. Парадигмы программирования. Общие принципы построения и использования языков программирования; средства описания данных; средства описания действий.
3. Теоретические проблемы разработки языков программирования. Стандарты языков программирования.
4. Алгоритм. Алгоритмы на графах. Минимальные покрывающие деревья. Алгоритмы Крускала и Прима.

5. Структуры данных. Двоичные деревья поиска. Красно-черные деревья.
6. Методы построения и анализа алгоритмов. Жадные алгоритмы. Задача о выборе заявок. Теоретические основы жадных алгоритмов.
7. Алгоритм. Формы представления. Способы представления и записи алгоритмов. Описание известных алгоритмов.
8. Алгоритмы сортировки одномерных массивов.
9. История возникновения программирования. Основные принципы и подходы при создании языков программирования.
10. Программирование. Стили программирования. История развития стилей программирования.
11. Системное программирование. История возникновения. Основные принципы, обзор основных процедур и функций.
12. Способы хранения информации в ПК. Представление целых, вещественных и текстовых форматов.
13. Логическое программирование. История возникновения. Основные принципы, обзор основных процедур и функций.
14. Функциональное программирование. История возникновения. Основные принципы, обзор основных процедур и функций.
15. Информация. Свойства информации. Информационные процессы: получение, передача, преобразование и хранение информации.
16. Объектно-ориентированное программирование. История развития. Иерархия объектов. Основные свойства объектно-ориентированного языка программирования.
17. Визуальные языки программирования. Обзор. История возникновения. Основные принципы, обзор основных процедур и функций.
18. Основы компьютерной графики. Основные понятия. Представление изображения. Разрешающая способность. Аппаратная поддержка графики.
19. Технология разработки программных продуктов. Этапы создания.
20. Моделирование потоков данных (процессов). Общие сведения. Состав диаграмм потоков данных. Построение иерархии диаграмм потоков данных.

Задания практической части курсовой работы

1. Разработка информационной системы в области машиностроения
2. Разработка информационной системы в области приборостроения
3. Разработка информационной системы в области науки
4. Разработка информационной системы в области техники
5. Разработка информационной системы в области образования
6. Разработка информационной системы в области медицины
7. Разработка информационной системы в области административного управления
8. Разработка информационной системы в области юриспруденции
9. Разработка информационной системы в области бизнеса
10. Разработка информационной системы в области предпринимательства
11. Разработка информационной системы в области коммерции
12. Разработка информационной системы в области менеджмента
13. Разработка информационной системы в области банковских систем
14. Разработка информационной системы в области безопасности информационных систем
15. Разработка информационной системы в области управления технологическими процессами
16. Разработка информационной системы в области механики
17. Разработка информационной системы в области технической физики
18. Разработка информационной системы в области энергетики
19. Разработка информационной системы в области ядерной энергетики

20. Разработка информационной системы в области силовой электроники
21. Разработка информационной системы в области металлургии
22. Разработка информационной системы в области строительства
23. Разработка информационной системы в области транспорта
24. Разработка информационной системы в области железнодорожного транспорта
25. Разработка информационной системы в области связи
26. Разработка информационной системы в области телекоммуникаций
27. Разработка информационной системы в области управления инфокоммуникациями
28. Разработка информационной системы в области почтовой связи
29. Разработка информационной системы в области химической промышленности
30. Разработка информационной системы в области сельского хозяйства
31. Разработка информационной системы в области текстильной и легкой промышленности
32. Разработка информационной системы в области пищевой промышленности
33. Разработка информационной системы в области медицинской и биотехнологической отрасли
34. Разработка информационной системы в области горного дела
35. Разработка информационной системы в области обеспечения безопасности подземных предприятий и производств
36. Разработка информационной системы в области геологии
37. Разработка информационной системы в области нефтегазовой отрасли
38. Разработка информационной системы в области геодезии и картографии
39. Разработка информационной системы в области геоинформационных систем
40. Разработка информационной системы в области лесного комплекса
41. Разработка информационной системы в области химико-лесного комплекса
42. Разработка информационной системы в области экологии
43. Разработка информационной системы в области сферы сервиса
44. Разработка информационной системы в области систем массовой информации
45. Разработка информационной системы в области дизайна
46. Разработка информационной системы в области медиаиндустрии
47. Разработка информационной системы для предприятия различного профиля в условиях экономики информационного общества

3.2 Показатели, критерии и шкала оценивания ответов на экзамене, подготовки курсовой работы

Критерии оценки курсовой работы

«Отлично» выставляется за курсовую работу, в которой:

1. Используется основная литература по проблеме.
2. Дано теоретическое обоснование актуальной темы и анализ передового опыта работы.
3. Показано применение научных методик и передового опыта в своей работе с испытуемыми, обобщен собственный опыт, иллюстрируемый различными наглядными материалами, сделаны выводы и даны практические рекомендации.
4. Работа безукоризненна в отношении оформления (орфография, стиль, цитаты, ссылки и т.д.).
5. Все этапы выполнены в срок.

«Хорошо» выставляется в случае, если:

1. Использована основная литература по теме (методическая и научная).
2. Дано теоретическое обоснование и анализ передового опыта работы.
3. Все этапы выполнены в срок.
4. Работа правильно оформлена.

5. Недостаточно описан личный опыт работы, применение научных исследований и передового опыта работы.

«Удовлетворительно» выставляется:

1. Библиография ограничена.
2. Нет должного анализа литературы по проблеме.
3. Хорошо обобщен, собственный опыт работы.
4. Оформление работы правильное.
5. Большая часть выполнена в срок.

Критерии оценивания	Показатели и шкала оценивания			
	5	4	3	2
полнота и правильность ответа	обучающийся полно излагает материал, дает правильное определение основных понятий	Обучающийся достаточно полно излагает материал, однако допускает 1-2 ошибки, которые сам же исправляет, и 1-2 недочета в последовательности и языковом	обучающийся демонстрирует знание и понимание основных положений данной темы, но излагает материал неполно и допускает	обучающийся демонстрирует незнание большей части соответствующего вопроса
степень осознанности, понимания изученного	демонстрирует понимание материала, может обосновать свои суждения, применить знания на практике, привести необходимые примеры не	присутствуют 1-2 недочета в обосновании своих суждений, количество приводимых примеров ограничено	не умеет достаточно глубоко и доказательно обосновать свои суждения и привести свои примеры	допускает ошибки в формулировке определений и правил, искажающие их смысл
языковое оформление ответа	излагает материал последовательно и правильно с точки зрения норм	излагает материал последовательно, с 2-3 ошибками в языковом оформлении	излагает материал непоследовательно и допускает много ошибок в языковом	беспорядочно и неуверенно излагает материал