



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.В.ДВ.5.1 «Языки 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:	2	2
Составитель:	Лапшина М.Л.	

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.....	4
1.3 Критерии оценивания результата обучения по дисциплине и шкала оценивания	5
2. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ	5
2.1 Текущий контроль	5
2.2 Критерии оценки качества освоения дисциплины.....	25
3. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ	27
3.1 Теоретические вопросы и практические задания для проведения экзамена.....	27
3.2 Показатели, критерии и шкала оценивания ответов на экзамене, подготовки курсовой работы.....	30

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-1	владение широкой общей подготовкой (базовыми знаниями) для решения практических задач в области информационных систем и технологий	знать: современные тенденции развития информатики и вычислительной техники, компьютерных технологий; уметь: применять вычислительную технику для решения практических задач; владеть: методами, способами и средствами работы с компьютером с целью получения, хранения и переработки информации.
ПК-11	способность к проектированию базовых и прикладных информационных технологий	Знать: основные принципы устройства информационных систем и сервисов. Уметь: выполнять информационный анализ инфокоммуникационных систем и сетей. Владеть: информационными технологиями для сопровождения информационных систем и сервисов.
ПК-23	готовностью участвовать в постановке и проведении экспериментальных исследований	знать: методологию определения целей и задач проведения экспериментальных исследований. уметь: проводить экспериментальные исследования, применять методы планирования экспериментов, анализировать результаты экспериментальных исследований. владеть: современными инструментальными средствами планирования экспериментов и анализа их результатов.
ПК-25	способность использовать математические методы обработки, анализа и синтеза результатов профессиональных исследований	знать: принципы моделирования, классификацию способов представления моделей систем; приемы, методы, способы формализации объектов, процессов, явлений и реализацию их на компьютере; достоинства и недостатки различных способов представления моделей систем; разработку алгоритмов фиксации и обработки результатов моделирования систем; способы планирования машинных экспериментов с моделями. уметь: использовать технологии моделирования; представлять модель в математическом и алгоритмическом виде; оценивать качество модели; показывать теоретические основания модели. владеть: инструментальными средствами

		построения имитационных моделей информационных процессов, получением концептуальных моделей систем, построением моделирующих алгоритмов.
--	--	--

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. Основные понятия языков программирования.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
2	Тема 2. Функции языка программирования	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
3	Тема 3. Объектно – ориентированное программирование.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
4	Тема 4. Формальные грамматики и языки.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
5	Тема 5. Динамические структуры данных.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
6	Тема 6. Основные принципы построения трансляторов.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
7	Тема 7. Построение таблицы идентификаторов.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
8.	Тема 8. Построение лексического анализатора.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
9.	Тема 9. Построение синтаксического анализатора.	ОПК-1, ПК-11, ПК-23, ПК 25	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая работа, экзамен
10.	Тема 10. Генерация кода	ОПК-1, ПК-11, ПК-23,	Опрос на лабораторном практикуме, вопросы для контроля знаний, задания для СР, тестирование, курсовая

		ПК 25	работа, экзамен
--	--	-------	-----------------

№ п/п	Наименование лабораторных работ
1.	Лабораторная работа 1. Программирование с использованием функций
2.	Лабораторная работа 2. Программирование с использованием шаблонов
3.	Лабораторная работа 3. Композиция классов и объектов.
4.	Лабораторная работа 4. Массивы, конструкторы и перегрузка операций.
5.	Лабораторная работа 5. Наследование, виртуальные методы.
6.	Лабораторная работа 6. Динамические структуры данных.
7.	Лабораторная работа 7. Проектирование таблицы идентификаторов.
8.	Лабораторная работа 8. Лексический анализатор.
9.	Лабораторная работа 9. Синтаксический анализатор.
10.	Лабораторная работа 10. Оптимизирующий генератор кода.

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) (превосходит пороговый (базовый) уровень по одному или нескольким существенным признакам)	студент дает ответ, удовлетворяющий тем же требованиям, что и для оценки «5», но допускает 1-2 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. Основные понятия языков программирования.

Контрольные вопросы:

1. Основные концепции языков программирования.
2. Классификация языков программирования.
3. Императивные языки.
4. Языки декларативного программирования.

Вопросы для контроля знаний:

1. Объектно – ориентированные языки.
2. Критерии языков программирования.

Лабораторная работа 1. Программирование с использованием функций

Цель работы: Освоить технологию программирования с использованием функций

Вопросы и задания по лабораторной работе 1

Ввести и обработать:

- 1) два двумерных массива, содержащие соответственно 3x5 и 4x8 вещественных элементов;
 - 2) три массива, содержащие соответственно 3, 6 и 8 целых элементов без знака;
 - 3) четыре массива, содержащие соответственно 4, 6, 3 и 5 целых элементов со знаком;
 - 4) два массива, содержащие соответственно 4 и 6 вещественных элементов;
 - 5) три массива, содержащие соответственно 5, 10 и 4 целых элементов без знака;
 - 6) четыре массива, содержащие соответственно 3, 5, 8 и 6 вещественных элементов;
 - 7) два трехмерных массива, содержащие соответственно 2x3x2 и 3x4x2 вещественных элементов;
 - 8) четыре массива, содержащие соответственно 4, 7, 3 и 5 вещественных элементов;
 - 9) три двумерных массива, содержащие соответственно 2x5, 3x6 и 3x4 целых элементов без знака;
 - 10) два двумерных массива, содержащие соответственно 6x2 и 3x2 вещественных элементов.
1. Для чего предназначены функции?
 2. Чем отличаются формальные и фактические параметры?
 3. Чем отличаются глобальные и локальные переменные?

Тема 2. Функции языка программирования

Контрольные вопросы:

1. Императивные языки.
2. Структурное программирование.
3. Функции языка.

Вопросы для контроля знаний

1. Передача параметров в функцию.
2. Побочный эффект.

Лабораторная работа 2. Программирование с использованием шаблонов

Цель работы: Освоить технологию программирования с использованием шаблонов

Вопросы и задания по лабораторной работе 2

1. В чем смысл использования шаблонов?
2. Синтаксис/семантика шаблонов функций?
3. Синтаксис/семантика шаблонов классов?
4. Определить параметризованную функцию сортировки массива методом обмена.

5. Определить шаблон класса “вектор” - одномерный массив.
 6. Что такое параметры шаблона функции?
 7. Перечислите основные свойства параметров шаблона функции.
 9. Можно ли перегружать параметризованные функции?
 10. Перечислите основные свойства параметризованных классов.
 11. Может ли быть пустым список параметров шаблона? Объясните.
 12. Как вызвать параметризованную функцию без параметров?
 13. Все ли компонентные функции параметризованного класса являются параметризованными?
 14. Являются ли дружественные функции, описанные в параметризованном классе, параметризованными?
 15. Могут ли шаблоны классов содержать виртуальные компонентные функции?
 16. Как определяются компонентные функции параметризованных классов вне определения шаблона класса?
 17. Каковы синтаксис/семантика “операции-функции”?
 18. Когда нужно перегружать операцию присваивания для определенного пользователем типа данных, например класса?
 19. Можно ли изменить приоритет перегруженной операции?
 20. Можно ли изменить количество операндов перегруженной операции?
 22. Можно ли, используя дружественную функцию, перегрузить оператор присваивания?
 23. Все ли операции языка могут быть перегружены?
 25. Все ли операции можно перегрузить с помощью глобальной дружественной функции?
 26. В каких случаях операцию можно перегрузить только глобальной функцией?
 27. В каких случаях глобальная операция-функция должна быть дружественной?
- Задания:** При тестировании созданных шаблонов классов необходимо создавать объекты с различными допустимыми значениями параметров шаблона (например, компоненты вектора могут быть целыми, действительными или комплексными числами).
1. Создать шаблон класса для работы со стеком.
 2. Создать шаблон класса для работы с одномерным массивом. Выполнить тестирование путем создания и обработки массивов, содержащих элементы различных типов (например, сортировка элементов массивов различными методами).
 3. Создать шаблон класса Vector размерности n
 4. Создать шаблон класса «Квадратная матрица» – Matrix размерности $n \times n$
 5. Создать шаблон класса Polynom степени n
 6. Создать шаблон класса для работы с односвязным списком.
 7. Создать шаблон класса для работы с односвязным списком.
 8. Создать шаблон класса для работы с бинарным деревом.
 9. Создать шаблон класса Set (множество) мощности n
 10. Создать шаблон класса для работы с односвязным списком.
 11. Создать шаблон класса, обеспечивающего описание матрицы заданного размера $n \times m$ и любого минора в ней
 12. Создать шаблон класса для работы с бинарным деревом.
 13. Создать шаблон класса для работы с бинарным деревом.
 14. Создать шаблон класса для работы с двусвязным списком.

Тема 3. Объектно – ориентированное программирование.

Контрольные вопросы:

1. Классы и объекты.
2. Конструкторы.

3. Деструкторы.
4. Статические поля и методы.
5. Наследование.

Вопросы для контроля знаний:

1. Дружественные классы и функции.
2. Наследование.
3. Виртуальные методы.
4. Перегрузка операций.

Лабораторная работа 3. Композиция классов и объектов.

Цель работы: Получить практические навыки реализации классов

Вопросы по лабораторной работе 3

1. Что такое класс?
2. Что такое объект?
3. Как связаны между собой классы и объекты в программе?
4. Что такое инкапсуляция?
5. За счет чего реализуется защита от несанкционированного доступа к данным?
6. Чем отличаются поля от переменных?
7. Что такое свойство?
8. Какие методы есть у свойства?
9. Что делают эти методы?
10. Что такое точка входа?

Тема 4. Формальные грамматики и языки.

Контрольные вопросы:

1. Языки и цепочки символов.
2. Операции над цепочками.
3. Понятие языка.
4. Формальное определение языка.
5. Грамматики.
6. Классификация грамматик.
7. Вывод.
8. Сентенциальная форма.

Вопросы для контроля знаний:

1. Дерево вывода.
2. Проблемы однозначности и эквивалентности грамматик.
3. Распознаватели и их классификация.

Лабораторная работа 4. Массивы, конструкторы и перегрузка операций.

Цель работы: изучить возможности программирования классов на языке и получить основные навыки программирования

Вопросы и задания по лабораторной работе 4

1. Какие классы и функции называются дружественными?
2. Как осуществляется перегрузка операций?
3. Сколько аргументов требуется для определения перегруженной унарной (бинарной) операции?

Тема 5. Динамические структуры данных.

Контрольные вопросы:

1. Линейные списки.
2. Стеки.
3. Использование стеков для проверки правильности расстановки скобок в выражении.
4. Очереди.

Вопросы для контроля знаний:

1. Применение бинарных деревьев для сортировки и поиска.
2. Циклические списки.
3. Задача Джозефуса.
4. Бинарные деревья

Лабораторная работа 5. Наследование, виртуальные методы.

Цель: Получить практические навыки создания иерархии классов и использования статических компонентов класса.

Вопросы и задания по лабораторной работе 5

1. Создать базовый класс Car(машина), характеризующийся торговой маркой (строка), числом цилиндров, мощностью. Определить методы переназначения и изменения мощности. Создать производный класс Loggy (грузовик), характеризующийся также грузоподъемностью кузова. Определить функции переназначения марки и изменения грузоподъемности.
2. Создать класс Pair(пара чисел); определить методы изменения полей и сравнения пар: $\text{par.p1} > \text{par.p2}$, если $(\text{first.p1} > \text{first.p2})$ или $(\text{first.p1} = \text{first.p2})$ и $(\text{second.p1} > \text{second.p2})$. Определить класс-наследник Fraction с полями: целая часть числа и дробная часть числа. Определить полный набор методов сравнения.
3. Создать класс Liquid (жидкость), имеющий поля названия и плотности. Определить методы переназначения и изменения плотности. Создать производный класс Alcohol(спирт), имеющий крепость. Определить методы переназначения и изменения крепости.
4. Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс Rectangle (прямоугольник) с полями-сторонами. Определить методы вычисления периметра и площади прямоугольника.
5. Создать класс Man (человек), с полями: имя, возраст, пол и вес. Определить методы переназначения имени, изменения возраста и изменения веса. Создать производный класс Student, имеющий поле года обучения. Определить методы переназначения и увеличения года обучения.
6. Создать класс Triad (тройка чисел); определить методы изменения полей и вычисления суммы чисел. Определить производный класс Triangle с полями-сторонами. Определить методы вычисления углов и площади треугольника.
7. Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс Equilateral (равносторонний), имеющий поле площади. Определите метод вычисления площади.
8. Создать класс Triangle с полями-сторонами. Определить методы изменения сторон, вычисления углов, вычисления периметра. Создать производный класс RightAngled (прямоугольный), имеющий поле площади. Определить метод вычисления площади.
9. Создать класс Pair (пара чисел); определить методы изменения полей и вычисления произведения чисел. Определить производный класс RightAngled с полями-катетами. Определить методы вычисления гипотенузы и площади треугольника.
10. Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Date с полями: год, месяц и день. Определить полный набор методов сравнения дат.
11. Создать класс Triad (тройка чисел); определить метод сравнения триад (см. задание 2). Определить производный класс Time с полями: час, минута и секунда. Определить полный набор методов сравнения моментов времени.

12. Реализовать класс-оболочку Number для числового типа float. Реализовать методы сложения и деления. Создать производный класс Real, в котором реализовать метод возведения в произвольную степень, и метод для вычисления логарифма числа.

13. Создать класс Triad (тройка чисел); определить методы увеличения полей на 1. Определить производный класс Date с полями: год, месяц и день. Переопределить методы увеличения полей на 1 и определить метод увеличения даты на n дней.

14. Реализовать класс-оболочку Number для числового типа double. Реализовать методы умножения и вычитания. Создать производный класс Real, в котором реализовать метод, вычисляющий корень произвольной степени, и метод для вычисления числа π в данной степени.

15. Создать класс Triad (тройка чисел); определить методы увеличения полей на 1. Определить класс-наследник Time с полями: час, минута, секунда. Переопределить методы увеличения полей на 1 и определить методы увеличения на n секунд и минут.

1. Определить иерархию классов (в соответствии с вариантом).

2. Определить в классе статическую компоненту - указатель на начало связанного списка объектов и статическую функцию для просмотра списка.

3. Реализовать классы.

4. Написать демонстрационную программу, в которой создаются объекты различных классов и помещаются в список, после чего список просматривается. 5. Сделать соответствующие методы не виртуальными и посмотреть, что будет.

6. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

Тема 6. Основные принципы построения трансляторов.

Контрольные вопросы:

1. Расскажите об известных Вам последних разработках языков программирования. Приведите основные характеристики названных языков.

2. Приведите конкретные примеры использования методов трансляции в областях, не связанных с языками программирования.

3. Приведите конкретные примеры компилируемых языков программирования.

4. Приведите конкретные примеры интерпретируемых языков программирования.

5. Приведите конкретные примеры языков программирования, для которых имеются как компиляторы, так и интерпретаторы.

6. Основные достоинства и недостатки компиляторов.

Вопросы для контроля знаний:

7. Основные достоинства и недостатки интерпретаторов.

8. Опишите основные различия в синтаксисе двух известных Вам языков программирования.

9. Опишите основные различия в семантике двух известных Вам языков программирования.

10. Назовите основные фазы процесса трансляции и их назначение.

11. Назовите специфические особенности однопроходной трансляции.

12. Назовите специфические особенности многопроходной трансляции.

13. Приведите примеры возможных комбинаций однопроходной и многопроходной трансляции. Расскажите о практическом использовании этих схем.

Лабораторная работа 6. Динамические структуры данных.

Цель: освоить понятия динамических структур данных

Вопросы по лабораторной работе 6

1. Манипулирование адресами области памяти с помощью указателей

2. Разработка и реализация программ с файлами данных
3. Общая схема работы транслятора.
4. Интерпретаторы.
5. Ассемблеры.
6. Понятие прохода.
7. Однопроходные и многопроходные компиляторы. Этапы трансляции.

Тема 7. Построение таблицы идентификаторов.

Контрольные вопросы:

1. Принципы организации таблицы идентификаторов.
2. Простейшие способы построения таблицы идентификаторов: простой и упорядоченный список, бинарное дерево.
3. Хэш – функция и хэш – адресация.
4. Рехэширование.

Вопросы для контроля знаний:

1. Метод цепочек.
2. Комбинированные методы построения таблицы идентификаторов.

Лабораторная работа 7. Проектирование таблицы идентификаторов.

Цель: Освоить технологию проектирование таблиц идентификаторов

Вопросы по лабораторной работе 7

1. Динамическое программирование
2. Разработка рекурсивных алгоритмов

Тема 8. Построение лексического анализатора.

Контрольные вопросы:

1. Регулярные языки и грамматики.
2. Конечные автоматы.
3. Детерминированные и недетерминированные конечные автоматы.
4. Минимизация конечных автоматов.

Вопросы для контроля знаний:

1. Регулярные множества и регулярные выражения.
2. Эквивалентность трех способов задания регулярных языков
3. Чем отличаются таблица лексем и таблица идентификаторов?
4. В какую из этих таблиц лексический анализатор не должен помещать ключевые слова, разделители и знаки операций?

Лабораторная работа 8. Лексический анализатор.

Цель: Научиться строить и создавать лексический анализатор

Вопросы по лабораторной работе 8

1. Реализация рекурсивных структур.
2. Применение рекурсивных алгоритмов для различных сортировок
3. Какую роль выполняет лексический анализ в процессе компиляции?
4. Как связаны лексический и синтаксический анализ?
5. Какие проблемы необходимо решить при построении лексического анализатора на основе конечного автомата?

Тема 9. Построение синтаксического анализатора.

Контрольные вопросы:

1. Контекстно - свободные грамматики и языки.
2. Автоматы с магазинной памятью.
3. Приведенные грамматики.
4. Синтаксические распознаватели с возвратом.
5. Нисходящие распознаватели.
6. Восходящие распознаватели

Вопросы для контроля знаний:

1. Синтаксические распознаватели на основе грамматик предшествования.
2. Грамматики операторного предшествования.

Лабораторная работа 9. Синтаксический анализатор.

Цель: Рассмотреть и научиться работать с синтаксическим анализатором

Вопросы по лабораторной работе 9

1. Модульная структура программы.
2. Отладка модулей
3. Какие существуют методы задания языков? Сравните их по эффективности.
4. Что такое грамматика языка?
5. Какие существуют формы описания грамматик?
6. Какие типы языков выделяются по классификации Хомского? Как они соотносятся между собой?
7. Что такое сентенциальная форма грамматики?
8. Что является предложением языка?

Тема 10. Генерация кода**Контрольные вопросы:**

1. Распределение памяти, его принципы.
2. Виды переменных и областей памяти.
3. Статическое и динамическое связывание.
4. Генерация кода, ее общие принципы.
5. Синтаксически управляемый перевод.
6. Способы внутреннего представления программы.

Вопросы для контроля знаний:

1. Многоадресный код с явно именуемым результатом –тетрады.
2. Многоадресный код с неявно именуемым результатом – триады.
3. Обратная польская инверсная запись операций.

Лабораторная работа 10. Оптимизирующий генератор кода.

Цель: Рассмотреть и освоить построение оптимизирующего генератора кода

Вопросы по лабораторной работе 10

1. Что такое транслятор, компилятор и интерпретатор?
2. Расскажите об общей структуре компилятора.
3. Как строится дерево вывода (синтаксического разбора)? Какие исходные данные необходимы для его построения?
4. Объясните работу алгоритма генерации объектного кода по дереву синтаксического разбора.
5. За счет чего обеспечивается возможность генерации кода на разных объектных языках по одному и тому же дереву?
6. Какую роль выполняет генерация объектного кода в процессе компиляции?

7. Какие данные необходимы компилятору для генерации объектного кода? Какие действия выполняет компилятор перед генерацией?
8. Дайте определение понятию оптимизации программы. Для чего используется оптимизация?
9. Какие существуют методы оптимизации объектного кода?
10. Что такое триады и для чего они используются? Какие еще существуют методы для представления объектных команд?
11. Объясните работу алгоритма свертки.
12. Что такое лишняя операция? Что такое «число зависимости»?
13. Объясните работу алгоритма исключения лишних операций.

Практические 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 $N \cdot 3$.
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 повторений.
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 то есть 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 «Языки программирования»

1. Языком программирования называется

- 1) совокупность средств и правил перевода текста с естественного языка на формальный.
- 2) совокупность средств и правил перевода текста с формального языка на естественный.
- 3) совокупность средств и правил представления алгоритма в виде, пригодном для выполнения вычислительной машиной.
- 4) язык, понятный программистам.
- 5) язык, понятный компьютеру.

2. Система программирования – это:

- 1) устройство для создания компьютерных программ.
- 2) специальная программа, предназначенная для создания компьютерных программ.
- 3) операционная система компьютера.
- 4) программное обеспечение компьютера, предназначенное для разработки, отладки и исполнения программ, записанных на определённом языке программирования.
- 5) совокупность программ на определённом языке программирования.

3. Константами называются

- 1) элементы данных, не имеющие значений.
- 2) элементы данных, обозначаемые словом const.
- 3) элементы данных, значения которых в процессе выполнения программы могут изменяться или не изменяться в зависимости от условия.
- 4) элементы данных, значения которых в процессе выполнения программы изменяются.
- 5) элементы данных, значения которых в процессе выполнения программы не изменяются.

4. В языке Turbo Pascal используются константы следующих видов:

- 1) числовые, логические, символьные и строковые.
- 2) постоянные и переменные.
- 3) числовые и буквенные.
- 4) логические и булевские.
- 5) символьные и строковые.

5. Символьные и строковые константы – это

- 1) буквы и строки.
 - 2) буквы и слова.
 - 3) цифры и буквы.
 - 4) цифры и строки букв.
 - 5) отдельные символы и их последовательности.
6. Тип данных определяет...
- 1) множество значений, которые могут принимать объекты программы.
 - 2) множество значений, допустимых для операций программы.
 - 3) множество значений, которые могут принимать объекты программы, а также совокупность операций, допустимых над этими значениями.
 - 4) совокупность операций, допустимых над этими значениями.
 - 5) совокупность операций, допустимых над объектами программы.
7. Целочисленные типы, символьный, логический и пользовательские типы данных образуют группу
- 1) подпорядковых типов.
 - 2) порядковых типов.
 - 3) порядочных типов.
 - 4) непорядочных типов.
 - 5) произвольных типов.
8. Слова языка программирования Turbo Pascal подразделяются на:
- 1) зарезервированные слова, стандартные идентификаторы и идентификаторы пользователя.
 - 2) зарезервированные слова и имена.
 - 3) идентификаторы и резервные слова.
 - 4) стандартные идентификаторы.
 - 5) идентификаторы пользователя.
9. Метод решения задачи, записанный по определённым правилам, обеспечивающим однозначность его понимания и механического исполнения при всех значениях исходных данных (из некоторого множества значений), называется...
- 1) планом.
 - 2) законом.
 - 3) блок-схемой.
 - 4) алгоритмом.
 - 5) программой.
10. Алгоритмы делятся на три основных типа:
- 1) линейные, разветвляющиеся и цикловые.
 - 2) линейные, разветвляющиеся и циклические.
 - 3) линейные, ветвические и циклические.
 - 4) прямолинейные, разветвляющиеся и циклические
 - 5) прямолинейные, разветвляющиеся и цикловые.
11. Блок-схема алгоритма представляет собой
- 1) систему произвольных фигур.
 - 2) систему различных геометрических фигур.
 - 3) систему различных стереометрических фигур.
 - 4) систему связанных геометрических фигур.
 - 5) систему несвязанных геометрических фигур.

12. Так называется этап, изображаемый этим блоком.

- 1) Прерывание.
- 2) Передача данных.
- 3) Процесс.
- 4) Принятие решения.
- 5) Модификация.

13. Данный блок обозначает:

- 1) начало и конец алгоритма.
- 2) ввод или вывод информации.
- 3) арифметический блок.
- 4) логический блок, проверяющий истинность или ложность некоторого условия.
- 5) итерационный блок.

14. Операторы, которые не содержат внутри себя других операторов, называются

- 1) составными.
- 2) сложными.
- 3) простыми.
- 4) пустыми.
- 5) ложными.

15. Такие операторы языка Turbo Pascal, как составной, оператор условного перехода, операторы цикла, оператор выбора, оператор присоединения в записях, относятся к

- 1) конструктивным операторам.
- 2) операционным операторам.
- 3) операторным данным.
- 4) простым операторам.
- 5) сложным операторам.

16. Процедуры write и writeln предусмотрены для

- 1) записи данных в памяти компьютера.
- 2) чтения данных из памяти компьютера.
- 3) ввода данных.
- 4) вывода данных.
- 5) замены данных.

17. Часть программы, в которой записывается последовательность выполняемых операторов, называется

- 1) заголовком.
- 2) описательной.
- 3) исполнительной.
- 4) текущей.
- 5) главной.

18 Это часть блок-схемы

- 1) линейного алгоритма.
- 2) разветвляющегося алгоритма.
- 3) циклического алгоритма.
- 4) вспомогательного алгоритма.
- 5) вложенного алгоритма.

19 Раздел описания констант начинается с зарезервированного слова

- 1) program.
- 2) begin.
- 3) array.
- 4) var.
- 5) const.

20 Целый тип переменных обозначается словом

- 1) real.
- 2) integer.
- 3) char.
- 4) boolean.
- 5) var.

21 Вещественный тип переменных языка Turbo Pascal обозначается словом

- 1) real.
- 2) integer.
- 3) char.
- 4) boolean.
- 5) var.

22 Операторные скобки в программе обозначаются словами

- 1) if и else.
- 2) writeln и readln.
- 3) program и end.
- 4) begin и end.
- 5) for и to.

23 В программе на Turbo Pascal алгебраическое выражение $(13b+53c+d):(7a+4)$ имеет вид:

- 1) $13*b+53*c+d / 7*a+4$.
- 2) $(13*b+53*c+d)/(7*a+4)$.
- 3) $(13b+53c+d)/(7a+4)$.
- 4) $(13*b+53*c+d):(7*a+4)$.
- 5) $(13b+53c+d):7a+4$.

24 В программе на Turbo Pascal алгебраическое выражение имеет вид:

- 1) $x^2/3$.
- 2) $x^{2/3}$.
- 3) корень(3; x^2).
- 4) $\ln(2*\exp(x)/3)$.
- 5) $\exp(2*\ln(x) / 3)$.

25 В программе на Turbo Pascal алгебраическое выражение имеет вид:

- 1) $\sqrt{x}$.
- 2) $\text{sqrt}(x)$.
- 3) корень(x).
- 4) $x^{(1/2)}$.

26 В программе на Turbo Pascal алгебраическое выражение $|x+y|$ имеет вид:

- 1) $\text{mod}(x+y)$.
- 2) $\text{abs}(x+y)$.
- 3) $\text{abc}(x+y)$.

- 4) $\text{abs}(x) + \text{abs}(y)$.
- 5) $\text{mod}(x) + \text{mod}(y)$.

27 Новый файл в среде Turbo Pascal можно создать при помощи команды меню

- 1) File – Open.
- 2) File – Exit.
- 3) File – Run.
- 4) File – New.
- 5) New – File.

28 Запуск программы на выполнение в среде Turbo Pascal осуществляется с помощью меню

- 1) Compile – Run.
- 2) File – Run.
- 3) Run – Run. +
- 4) Run – Begin.
- 5) Debug – Run.

29 Открыть файл в среде Turbo Pascal можно при помощи команды меню

- 1) File – Open.
- 2) File – Exit.
- 3) File – Run.
- 4) File – New.
- 5) New File.

30 Результаты работы программы в среде Turbo Pascal можно посмотреть при помощи меню

- 1) Compile – Compile.
- 2) Run – Run.
- 3) File – Open.
- 4) Window – Next.
- 5) Debug – Output.

31 Для выхода из среды Turbo Pascal нужно использовать пункт меню

- 1) File – Save.
- 2) File – Exit. +
- 3) File – Open.
- 4) File – End.
- 5) File – Run.

32 Цикл представляет собой

- 1) произвольную последовательность операторов.
- 2) линейную последовательность операторов.
- 3) ветвящуюся последовательность операторов.
- 4) последовательность операторов, которая выполняется однократно.
- 5) последовательность операторов, которая выполняется неоднократно.

33 Приведённая схема отображает:

- 1) линейную последовательность.
- 2) ветвление.
- 3) цикл с параметром.
- 4) цикл с постусловием.
- 5) цикл с предусловием.

34 Здесь приведён пример программы

if Условие then

begin {эти инструкции выполняются, если Условие истинно}

end

else

begin {эти инструкции выполняются, если Условие ложно}

end;

1) линейного алгоритма.

2) разветвляющегося алгоритма.

3) циклического алгоритма.

4) вспомогательного алгоритма.

5) вложенного алгоритма.

35 Так называется цикл, который может быть представлен в следующей форме:

for p:= S1 to S2 do A (S1<S2).

1) цикл с увеличением счётчика.

2) цикл с уменьшением счётчика.

3) цикл с постусловием.

4) цикл с предусловием.

5) цикл с телом.

36 Так называется цикл, который в программе имеет вид: repeat – заголовок p1; p2; ...; тело цикла pk until B; – условие окончания цикла,

1) цикл с телом.

2) цикл с увеличением счётчика.

3) цикл с уменьшением счётчика.

4) цикл с постусловием.

5) цикл с предусловием.

37 В записи какой из констант допущена ошибка?

1) 25

2) 183E-03

3) 3,14+

4) 256

5) –10000

38 Так называется фиксированное количество упорядоченных однотипных компонент, снабжённых индексами.

1) файл.

2) каталог.

3) алгоритм.

4) программа.

5) массив.

39 Массив представляет собой

1) совокупность данных одного типа с общим для всех элементов именем.

2) совокупность данных одного типа с различным для всех элементов именем.

3) совокупность данных любого типа с общим для всех элементов именем.

4) совокупность данных любого типа с различным для всех элементов именем.

5) совокупность данных с именем.

40 Диапазон изменения индекса (индексов) массива определяет:

- 1) количество индексов у элемента.
- 2) количество элементов в массиве.
- 3) тип элементов массива.
- 4) тип переменных, описывающих индексы элементов.
- 5) номера элементов массива.

41 Максимальный объём памяти, который может занимать массив, в Турбо Паскале составляет

- 1) 6 байт.
- 2) 65 байт.
- 3) 655 байт.
- 4) 6552 байта.
- 5) 65520 байт.

42 Приведённый пример позволяет описать: `Var a: array[0..3] of real;`

- 1) линейный вычислительный процесс.
- 2) ветвящийся вычислительный процесс.
- 3) циклический вычислительный процесс.
- 4) одномерный массив.
- 5) двумерный массив.

43. Приведённый пример позволяет описать: `Const K1=1; K2=3; L1=5; L2=7; Type mas=array[K1..K2, L1..L2] of real; Var a:mas;`

- 1) линейный вычислительный процесс.
- 2) ветвящийся вычислительный процесс.
- 3) циклический вычислительный процесс.
- 4) одномерный массив.
- 5) двумерный массив.

44 Выберите фрагмент программы, позволяющий заполнить одномерный массив случайными числами в диапазоне от 0 до 99.

- 1) `Randomize; for i:=1 to n do a[i]:=random (100);`
- 2) `For i:=1 to 5 do readln (a[i]);`
- 3) `For i:=1 to 3 do for j:=1 to 2 do readln (a[i,j]);`
- 4) `For i:=1 to 5 do writeln (a[i]);`
- 5) `For i:=1 to 3 do for j:=1 to 2 do writeln (a[i,j]);`

45 Так называется процесс упорядочивания набора данных одного типа по возрастанию или убыванию значения какого-либо признака.

- 1) Замена.
- 2) Сортировка.
- 3) Сравнение.
- 4) Перестановка.
- 5) Поиск.

46 Приведённый пример программы демонстрирует, как можно: `Program massiv; Const n=10; Var a:array[1..n] of integer; s,i: integer; Begin Writeln('Исходный массив:'); For i:=1 to n do begin a[i]:=random(10); Write('a[' ,i, ']=',a[i], ' '); Writeln ( ' '); end; s:=1; For i:=1 to n do s:=s*a[i]; Writeln(S=' ,s); end.`

- 1) заполнить массив данными.
- 2) заполнить массив данными и найти максимальный элемент массива.
- 3) заполнить массив данными и вычислить сумму элементов массива.

- 4) заполнить массив данными и вычислить произведение элементов массива.
- 5) удалить значения элементов массива.

47 Приведённый фрагмент программы демонстрирует, как производится: For j:=1 to m do begin s:=0; for i:=1 to n do s:=s+a[i,j]; d[j]:=s; end;

- 1) суммирование всех элементов массива.
- 2) суммирование элементов каждой строки массива.
- 3) суммирование элементов каждого столбца массива.
- 4) суммирование элементов двух массивов.
- 5) вычисление произведения элементов массива.

48 Приведённый фрагмент программы демонстрирует, как производится: m:=a[1]; k:=1; for i:=2 to n do if a[i]>m then

begin

m:=a[i]; k:=i;

end;

- 1) поиск максимального элемента массива и его номера.
- 2) только поиск максимального элемента массива.
- 3) поиск минимального элемента массива и его номера.
- 4) только поиск минимального элемента массива.
- 5) поиск элемента массива, имеющего значение, равное 1.

49 Приведённый фрагмент программы демонстрирует, как производится:

C:=a[1,1]; K:=1; L:=1;

for i:=1 to n do

for j:=1 to m do if a[i,j]

begin

C:=a[i,j]; K:=i; L:=j;

end;

- 1) поиск максимального элемента массива.
- 2) поиск минимального элемента массива.
- 3) ввод элементов массива.
- 4) вывод всех элементов массива.
- 5) заполнение массива данными.

50 Для проверки чётности элементов массива можно использовать именно эту функцию, которая возвращает True, если число K – нечётное и False, если число K – чётное.

- 1) Chet(K).
- 2) Nechet(K).
- 3) Doo(K).
- 4) Odd(K).+
- 5) Mod(K).

51. Приведённый фрагмент программы демонстрирует описание файла именно этого вида: type filetype=text; var f: filetype;

- 1) Графический.
- 2) Числовой.
- 3) Типизированный.
- 4) Нетипизированный.
- 5) Текстовый.

52 Вид файла определяет в программе:

- 1) применяемые символы.
- 2) применяемые процедуры.
- 3) применяемы операторы.
- 4) способ хранения информации в файле.
- 5) внешний вид электронной страницы.

53 Файловые переменные Input и Output являются

- 1) логическими.
- 2) физическими.
- 3) текстовыми.
- 4) типизированными.
- 5) нетипизированными.

54 Файл, к элементам которого доступ выполняется в той же последовательности, в какой они записывались, называется

- 1) файлом последовательного доступа.
- 2) файлом прямого доступа.
- 3) файлом элементарного доступа.
- 4) файлом промежуточного доступа.
- 5) файлом доступа.

55 Процедура `reset(ФайловаяПеременная)` выполняет именно это действие.

- 1) Связывает файловую переменную с именем файла.
- 2) Открывает файл, связанный с файловой переменной для чтения, начиная с первого символа.
- 3) Открывает файл для записи в начало файла.
- 4) Открывает файл для записи в конец файла.
- 5) Удаляет неоткрытый внешний файл любого типа, задаваемый параметром `ФайловаяПеременная`.

56 Эта процедура создаёт и открывает новый файл для последующей записи данных.

- 1) `assign(ФайловаяПеременная, ИмяФайла)`
- 2) `reset(ФайловаяПеременная)`
- 3) `rewrite(ФайловаяПеременная)`
- 4) `rename(ФайловаяПеременная, ИмяФайла)`
- 5) `erase(ФайловаяПеременная)`

57 Так называется программная единица, имеющая имя, по которому она может быть вызвана из других частей программы.

- 1) Файл.
- 2) Каталог.
- 3) Программа.
- 4) Подпрограмма.
- 5) Оператор.

58 Процедура `append(ФайловаяПеременная)` выполняет именно это действие.

- 1) Открывает файл для записи в начало файла.
- 2) Открывает файл для записи в конец файла.
- 3) Удаляет неоткрытый внешний файл любого типа, задаваемый параметром `ФайловаяПеременная`.
- 4) Связывает файловую переменную с именем файла.

5) Открывает файл, связанный с файловой переменной для чтения, начиная с первого символа.

59 Процедура `erase(ФайловаяПеременная)` выполняет именно это действие

- 1) Связывает файловую переменную с именем файла.
- 2) Открывает файл, связанный с файловой переменной для чтения, начиная с первого символа.
- 3) Открывает файл для записи в начало файла.
- 4) Открывает файл для записи в конец файла.
- 5) Удаляет неоткрытый внешний файл любого типа, задаваемый параметром `ФайловаяПеременная`.

60 Функция `seekeof(ФайловаяПеременная)` проверяет

- 1) существование файла на диске.
- 2) не достигнуто ли начало файла при чтении из него данных.
- 3) не достигнуто ли начало строки текстового файла.
- 4) не достигнут ли конец текстового файла.
- 5) не достигнут ли конец строки текстового файла.

61 Выберите процедуру, которая производит запись в файл без перевода строки.

- 1) `read(ФайловаяПеременная, СписокПеременных)`.
- 2) `readln(ФайловаяПеременная, СписокПеременных)`.
- 3) `write(ФайловаяПеременная, СписокПеременных)`.
- 4) `writeln(ФайловаяПеременная, СписокПеременных)`.
- 5) `close(ФайловаяПеременная)`.

62 Так называется последовательность символов `char`, сгруппированных в строки, заканчивающиеся специальным символом `eoln`.

- 1) Текстовый файл
- 2) Типизированный файл.
- 3) Нетипизированный файл.
- 4) Физический файл.
- 5) Логический файл.

63 Этот тип файлов в программе объявляется следующим образом:

`var ФайловаяПеременная = text;`

- 1) Физический файл.
- 2) Логический файл.
- 3) Типизированный файл.
- 4) Нетипизированный файл.
- 5) Текстовый файл.

64 Этот тип файлов в программе объявляется следующим образом: `type ИмяТипа = file;`

`var ФайловаяПеременная : ИмяТипа;`

- 1) Физический файл.
- 2) Логический файл.
- 3) Типизированный файл.
- 4) Нетипизированный файл.
- 5) Текстовый файл.

65 Выберите пункт, который не описывает достоинства подпрограмм.

- 1) Программы, написанные с участием подпрограмм, легче тестировать и отлаживать, у них более чёткая логическая структура.
- 2) Подпрограммы делятся на два вида – процедуры и функции.
- 3) Независимость подпрограмм позволяет локализовать в них все детали программной реализации того или иного алгоритма, и поэтому их изменение обычно не приводит к изменению основной программы.
- 4) Самостоятельный характер подпрограмм позволяет поручать их составление различным программистам.
- 5) Использование подпрограмм позволяет экономить память.

66. Так называется независимая именованная часть программы, которую после однократного описания можно многократно вызывать по имени из последующих частей программы для выполнения определённых действий.

- 1) Файл.
- 2) Каталог.
- 3) Процедура.
- 4) Программа.
- 5) Оператор.

67 Пусть в разделе описаний описана функция: `Function F(x:integer):real;`

А также описаны некоторые переменные: `Var y,v,n: real; s,z:integer;`

Укажите вариант верного примера, который описывает вызов этой функции в основной программе.

- 1) `v:=F(z);`
- 2) `c:=F(z);`
- 3) `z:=F(s);`
- 4) `z:=F(y);`
- 5) `n:=F(v);`

68 Заголовок функции в программе описывается следующим образом:

- 1) `Function;`
- 2) `Function ИмяФункции;`
- 3) `Function ИмяФункции (ФормальныеПараметры);`
- 4) `Function ИмяФункции (ФормальныеПараметры): ТипРезультата;`
- 5) `Program ИмяФункции;`

69 Переменные, описанные в основной программе, являются по отношению к внутренним процедурам и функциям...

- 1) локальными.
- 2) глобальными.
- 3) формальными.
- 4) фактическими.
- 5) внутренними.

70 У формальных и фактических параметров должны совпадать:

- 1) только количество.
- 2) количество и тип.
- 3) количество и порядок следования.
- 4) порядок следования и тип.
- 5) количество, порядок следования, тип.

71 Так называются параметры, перед которыми в описании формальных параметров не ставится служебное слово var, и в процедуре работают только значения этих параметров.

- 1) Фактические параметры.
- 2) Формальные параметры.
- 3) Параметры-переменные.
- 4) Параметры-значения.
- 5) Параметры-процедуры.

72. В приведённом фрагменте программы a и b являются...

procedure hline(a:integer; b:char);

- 1) фактическими параметрами.
- 2) формальными параметрами.
- 3) параметрами-переменными.
- 4) параметрами-функциями.
- 5) параметрами-процедурами.

73 Так в программе описывается: Type ИмяТипа = Function(p1,p2,...,pn):ТипРезультата

- 1) параметр-функция.
- 2) параметр-процедура.
- 3) параметр-переменная.
- 4) параметр-значение.
- 5) фактический или формальный параметр.

74 Рассмотрите фрагмент программы: procedure EX(k,l:integer;var m:real);
В списке формальных параметров процедуры EX m представляет собой...

- 1) параметр-функцию.
- 2) параметр-процедуру.
- 3) параметр-переменную.
- 4) параметр-значение.
- 5) параметр.

75. Рассмотрите фрагмент программы: procedure QQ(k,l:integer;var m:real);
В списке формальных параметров процедуры QQ k представляет собой...

- 1) параметр.
- 2) параметр-процедуру.
- 3) параметр-переменную.
- 4) параметр-функцию.
- 5) параметр-значение.

2.2 Критерии оценки качества освоения дисциплины

Качество освоения дисциплины оценивается по степени успешности качества 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.

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.

Примерные задания практической части курсовой работы

1. Игра "Расстановка 16 букв". В квадрате размером 4 x 4 клетки расставить 16 букв (по четыре буквы a, b, c, d) так, чтобы в каждом горизонтальном и в каждом вертикальном ряду любая буква встречалась только один раз.
2. Игра "100 спичек". Из кучки, первоначально содержащей 100 спичек, двое играющих поочередно берут по несколько спичек: не менее одной и не более десяти. Выигрывает взявший последнюю спичку.
3. Игра "Вращающийся квадрат". Дан квадрат размером 4 x 4 клетки, в которых с помощью датчика случайных чисел расставлены буквы от А до Р. Упорядочить буквы в квадрате по алфавиту. Квадрат имеет подквадраты, которые можно вращать по часовой стрелке на одну клетку. Подквадраты имеют размер 2 x 2 и указываются номером левой верхней клетки.
4. Составить программу для обучения переводу чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную и обратно. Программа должна предлагать двоичное число, выбранное с помощью датчика случайных чисел, обучающийся - назвать число в восьмеричной и шестнадцатеричной системе счисления.
5. Игра "Загадки". На экран выдается загадка, которую надо отгадать.
6. Игра «Тренировка памяти». Требуется воспроизводить все усложняющийся порядок нажатия кнопок или рисунков.
7. Игра "Ним". Имеется три кучки спичек. Двое играющих по очереди делают ходы. Каждый ход заключается в том, что из одной какой-то кучки берется произвольное ненулевое число спичек. Проигрывает взявший последнюю спичку.
8. Игра "Мост". Дан мост с арками разной ширины, в нижней строке экрана расположен мяч, которым можно управлять: мяч можно перемещать по строке, останавливать в нужной позиции строки и катить к мосту. Очки начисляются, если мяч проходит через арку.
9. Составить программу для заучивания слов иностранного языка. Программа должна предлагать слова из некоторого списка на одном языке, обучающий - дать перевод этого слова на другой язык.
10. Игра "Быки и коровы". Программа случайно выбирает четырехзначное число. Угадать это число. На каждом шаге играющий называет 4-х значное число, программа сообщает об угаданных цифрах (быки) и стоящих на месте (коровы). Например, задано число 6482, число 5428 содержит 1 "корову" и 2 "быка".
11. Приложение «Простые методы сортировки: метод «пузырька», сортировка вставками, сортировка бинарными вставками, сортировка выбором».
12. Программа «Хеширование открытой адресацией».
13. Разработать программу «Орфография», которая проверяла бы правильность написания слов в некоторой строке, сравнивая их со словами из словаря. Использовать при этом в качестве словаря одномерный массив слов.
14. Разработать программу «Операторы». Считывается файл, написанный на языке высокого уровня (C#, C++, Pascal, JAVA). Найти все операторы присваивания.

15. Разработать программу «Символы». Во входном файле, написанном на языке высокого уровня (C#, C++, Pascal, JAVA), считается число символов { и } и выводится на печать сообщение об ошибке, если они не равны между собой или не сбалансированы.

16. Разработать программу «Шифрование», которая предназначена для зашифровки текстов. Вывести на печать исходный текст и результат шифровки. Текст читать построчно, хранить текст в массиве строк.

17. Разработать программу «Строки», которая читает произвольные строки длиной до 80 символов, разбивает их по словам (подстрока между двумя пробелами), находит максимальное слово и выводит его на печать.

18. Работать программой, которая считывает произвольный текст и распечатывает в алфавитном порядке все латинские буквы, входящие в этот текст.

19. Разработать программу «Алфавит», которая считывает произвольный текст и распечатывает в алфавитном порядке все слова, входящие в этот текст.

20. Написать программу построения биоритмов по данному дню рождения на заданный интервал времени.

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 оформлении излагаемого	обучающийся демонстрирует знание и понимание основных положений данной темы, но излагает материал неполно и допускает неточности в определении понятий или формулировке правил	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 только из учебника, но и самостоятельно составленные	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 литературного языка	излагает материал последовательно, с 2-3 ошибками в языковом оформлении	излагает материал непоследовательно и допускает много ошибок в языковом оформлении излагаемого	беспорядочно и неуверенно излагает материал