

**КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)
АВТОНОМНОЙ НЕКОММЕРЧЕСКОЙ ОБРАЗОВАТЕЛЬНОЙ
ОРГАНИЗАЦИИ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»**

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Специальность: 10.05.01 Компьютерная безопасность

Специализация: «Разработка защищенного программного обеспечения»

Формы обучения: очная

Объем дисциплины:

в зачетных единицах: 5 з.е.

в академических часах: 180 ак.ч.

Форма промежуточной аттестации: экзамен

Оценочные материалы по дисциплине Основы программирования

обсуждены и рекомендованы к утверждению решением кафедры естественных дисциплин, сервиса и туризма от 25 марта 2025 г., протокол № 6

одобрены Научно-методическим советом Казанского кооперативного института (филиала) от 2 апреля 2025 г., протокол № 3

СОДЕРЖАНИЕ

1. Паспорт оценочных материалов по дисциплине
2. Оценочные материалы по дисциплине:
 - 2.1. Оценочные материалы для проведения текущего контроля.
 - 2.2. Оценочные материалы для проведения промежуточной аттестации.

Обновление оценочных материалов по дисциплине

Целью оценочных материалов по дисциплине (модулю) (далее –ОМ) является комплексная оценка результатов обучения по дисциплине Основы программирования и установление уровня сформированности компетенций обучающегося.

ОМ предназначен для проведения текущего контроля успеваемости и промежуточной аттестации.

ОМ включает оценочные средства для проведения текущего контроля успеваемости и промежуточной аттестации.

Оценочные материалы разработаны в соответствии с рабочей программой дисциплины Основы программирования.

1. Паспорт оценочных материалов по дисциплине «Основы программирования»

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
ОПК-7 Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ОПК-7.1. Демонстрирует умение создавать программы на языках высокого уровня, при менять методы и инструментальные средства структурного и объектно-ориентированного программирования для решения профессиональных задач	Знать: Типовые компьютерные методы и современное программное обеспечение для решения задач профессиональной деятельности на языке программирования высокого уровня. Уметь: Установить и использовать программное обеспечение для решения задач профессиональной деятельности. Применять методы компьютерного проектирования, разработки структур данных с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий на языке высокого	Тема 1. Алгоритм и его свойства. Алгоритм и его свойства. Машинный язык. Трансляция. Компиляторы и интерпретатор Тема 2. Понятие сегмента и регистра. Структура ЦП. Понятие сегмента и регистра. Структура ЦП. Линии управления т передачи данных. Тема 3. Функции языков программирования. Функции языков программирования. Основные свойства языков программирования. Основные аспекты изучения языков программирования Тема 4. Метаязыки описания языков программирования Метаязыки описания языков программирования. язык Бэкуса-Наура, язык синтаксических	Реферат (Тема 1-12) Тест (тема 1-12)	Тест (В 1-10 тестового задания)

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
		уровня. Владеть: Проектирование структур данных при реализации алгоритмических и программных решений на языке программирования высокого уровня.	диаграмм Тема 5. Виртуальная машина. Виртуальная машина. Типы ошибок, распознаваемые ВМ. Синтаксические и семантические ошибки. Тестирование. Тема 6. Структура типов языка C++ Структура программы процедурных языков. Понятие типа. Структура типов языка C++. Встроенные и конструируемые типы, скалярные типы. Тема 7. Целочисленные типы. Символьный тип Целочисленные типы. Символьный тип. Диапазон значений, набор операций. Тема 8. Булевский (логический) тип. Вещественные типы. Булевский (логический) тип. Вещественные типы. Диапазон значений, набор операций. Тема 9. Описание переменных и констант. Арифметические		

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
			операции. Описание переменных и констант. Арифметические операции. Условная операция. Выражения. Тема 10. Поразрядные операции языка C++. Операторы языка. Поразрядные операции языка C++. Примеры. Операторы языка. Оператор присваивания. Тема 11. Средства ввода-вывода. Условный оператор. Средства ввода-вывода. Условный оператор. Вложенность условных операторов. Тема 12. Оператор выбора. Операторы цикла. Оператор выбора. Операторы цикла for, while, do while. Условия выхода из цикла		
	ОПК-7.2. Демонстрирует умения разработки алгоритмов решения профессиональных	Знать: Создание и интеграция программных модулей и компонент программного продукта с использованием языка программирования C++. Типовые алгоритмические и	Тема 13. Операторы перехода. Обработка последовательностей. Операторы перехода. Обработка последовательностей. Программирование арифметических циклов.	Реферат (Тема 13-24) Тест (Тема 13-24) Практическое задание	Тест (В 11-25 тестового задания)

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
	за дач со сложной организацией структуры данных, анализа сложности алгоритмов, выбора оптимальных структур данных и реализации типовых алгоритмов и структур данных	программные решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения на языке C++. Основные алгоритмические и программные решения, методы и средства проектирования программного обеспечения разработанного на языке C++. Уметь: Разрабатывать и интегрировать программные модули и компоненты в единый программный продукт с использованием языка программирования C++ в среде Microsoft Visual Studio. Использовать существующие алгоритмические и программные решения и шаблоны проектирования программного обеспечения на языке C++. Применять методы и средства	Тема 14. Итерационные циклы. Программирование вложенных Циклов. Обработка последовательностей. Итерационные циклы. Программирование вложенных циклов. Тема 15. Регулярные типы. Обработка одномерных массивов. Регулярные типы. Определение массивов. Обработка одномерных массивов. Тема 16. Классы задач по обработке массивов. Задачи 1-ого и 2-го класса. Классы задач по обработке массивов. Задачи 1-ого класса. Обоснование выбора структуры данных. Классы задач по обработке массивов. Задачи 2-ого класса. Тема 17. Классы задач по обработке массивов. Задачи 3-ого и 4-го класса. Классы задач по обработке	(Тема 1-24)	

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
		проектирования программного обеспечения, структур данных, интерфейсов с использованием основных алгоритмических и программных решений в области информационно-коммуникационных технологий на языке C++ в среде Microsoft Visual Studio. Владеть: Разработка и интегрирование программных модулей и компонент в единый программный продукт с использованием языка программирования C++ в среде Microsoft Visual Studio. Проектирование структур данных при реализации алгоритмических и программных решений в среде Microsoft Visual Studio. Проектирование программных продуктов при реализации алгоритмических и программных решений в среде	массивов. Задачи 3- ого класса. Классы задач по обработке массивов. Задачи 4-ого класса. Тема 18. Двумерные массивы. Двумерные массивы. Обработка диагоналей. Построение массива по матрице. Тема 19. Функции. Прототип функции. Параметры функции. Функции. Прототип функции. Параметры функции. Передача параметров по значению. Передача параметров по адресу (по ссылке и через указатель). Тема 20. Локальные и глобальные переменные. Область видимости переменных. Локальные и глобальные переменные. Область видимости переменных. Параметры со значениями по умолчанию (необязательные параметры). Перегрузка функций. Тема 21. Рекурсия. Шаблоны функций.		

Формируемые компетенции (код и наименование компетенции)	Индикаторы достижения компетенций	Планируемые результаты обучения	Контролируемые разделы, темы дисциплины	Наименование оценочного средства ¹	
				Текущий контроль	Промежуточная аттестация
1	2	3	4	5	6
		Microsoft Visual Studio с использованием конструкций языка C++.	Рекурсия. Прямая и косвенная рекурсия. Стек активаций. Шаблоны функций. Тема 22. Методы сортировки. Методы сортировки. Общая постановка задачи. Оценивание алгоритма. Метод поиска с обменом (сортировка посредством выбора). Алгоритм «Пузырька». Челночная сортировка (сортировка вставками). Метод подсчета. Метод парных сравнений. Быстрая сортировка. Тема 23. Указатели. Динамические переменные. Указатели. Операции с указателями. Динамические переменные. Операторы new и dispose. Тема 24. Динамические массивы. Динамические массивы. Передача массивов как параметров в функцию.		

2. Оценочные материалы по дисциплине «Основы программирования»

2.1 Оценочные материалы для проведения текущего контроля успеваемости

Тематика рефератов

Тема 1. Алгоритм и его свойства

1. Основные свойства алгоритмов и критерии качества их реализации
2. Классификация алгоритмов по типу решаемых задач
3. Методы оценки эффективности алгоритмов
4. История развития понятия алгоритма и его значение в науке и технике
5. Алгоритмы сортировки: сравнительный анализ популярных методов
6. Применение рекурсивных алгоритмов в современной информатике
7. Разработка и реализация параллельных алгоритмов
8. Анализ базовых структур данных и их влияние на эффективность алгоритмов
9. Генетические алгоритмы и эволюционное программирование
10. Проблема NP-полных задач и пути её решения
11. Динамическое программирование: основные принципы и области применения
12. Защита данных методами криптографических алгоритмов
13. Использование вероятностных алгоритмов в анализе больших данных
14. Практическое использование жадных алгоритмов
15. Реализация алгоритмов машинного обучения на примере классификации и регрессии

Тема 2. Понятие сегмента и регистра. Структура ЦП..

1. Определение и назначение регистров процессора.
2. Архитектура сегментированной памяти и принцип сегментации.
3. Структурные особенности регистров общего назначения.
4. Регистры специальных функций (SFR): назначение и роль в управлении системой.
5. Организация и функционирование стековых регистров.
6. Механизм адресации сегментов и страниц в x86 архитектуре.
7. Современные тенденции в развитии архитектур регистровой модели процессоров.
8. Принцип конвейерной обработки данных и его связь с использованием регистров.
9. Особенности регистровых файлов в высокопроизводительных системах.
10. Устройство и работа блока целочисленных регистров.
11. Регистр флагов и его влияние на выполнение программы.
12. Эффективность использования регистров с точки зрения оптимизации компиляторов.
13. Процессорное ядро и структура его компонентов.
14. Методология работы шин данных и управления в многоядерных архитектурах.
15. Принципы организации и функционирования кэша L1/L2/L3.

Тема 3. Функции языков программирования.

1. Понятие функции в языках программирования и её виды.
2. Параметры и аргументы функций: передача значений и ссылок.
3. Локальные и глобальные переменные внутри функций.
4. Рекурсия и её использование в функциональном программировании.
5. Разбор и сравнение лямбда-выражений в Python и JavaScript.
6. Встроенные и пользовательские функции в Python.
7. Область видимости и замыкания в функциях JavaScript.
8. Перегрузка функций и операторов в C++.
9. Именованные и позиционные аргументы в функциях Python.
10. Генераторы и корутины в Python.
11. Функциональное программирование на JavaScript: карта, фильтр, свёртка.
12. Передача функций как аргументов другим функциям (callback-функции).
13. Декораторы и аннотации функций в Python.
14. Оптимизация функций с помощью мемоизации и кеширования результатов.
15. Функции высшего порядка и композиция функций в функциональных языках программирования.

Тема 4. Метаязыки описания языков программирования

1. Общая характеристика метаязыков и их назначение.

2. Формальное описание синтаксиса и семантики языков программирования.
3. Теория формальных грамматик и их роль в описании языков программирования.
4. BNF-нотация и EBNF-нотация: сравнение и примеры использования.
5. Графовые грамматики и их применение в метаязыках.
6. Специализированные языки разметки для метаданных (XML, JSON).
7. Контекстно-зависимая и контекстно-свободная грамматики в метаязыках.
8. Семантический анализ и абстрактные синтаксические деревья.
9. Язык Yacc/Bison и Lex/Flex: инструменты для генерации парсеров.
10. Основы атрибутивных грамматик и их практическое применение.
11. Программные средства автоматизации синтаксического анализа.
12. Преимущества и недостатки подхода на основе регулярных выражений.
13. Общие стандарты метаязыков для объектно-ориентированных языков.
14. Применение метаязыков в проектировании предметно-ориентированных языков (DSL).
15. Современные тенденции и перспективы развития метаязыков описания языков программирования.

Тема 5. Виртуальная машина.

1. Понятие виртуальной машины и её классификация.
2. Виртуализация аппаратных ресурсов компьютера.
3. История развития технологий виртуализации.
4. Основные компоненты виртуальной машины: гипервизор и гостевые ОС.
5. Аппаратная поддержка виртуализации: Intel VT-x и AMD-V.
6. Технология контейнеризации и отличие от полноценной виртуализации.
7. VMware Workstation: организация и управление виртуальными машинами.
8. Oracle VirtualBox: особенности и сферы применения.
9. Microsoft Hyper-V: встроенная технология виртуализации Windows Server.
10. Vagrant: автоматизация развёртывания виртуальных сред.
11. Docker: лёгкая виртуализация и микросервисная архитектура.
12. Пользовательские режимы виртуализационных платформ: QEMU/KVM.
13. Совмещение физической и виртуальной инфраструктуры в корпоративных сетях.
14. Безопасность виртуальных машин и защита виртуализованных систем.
15. Будущие направления развития виртуальных машин и облачных сервисов.

Тема 6. Структура типов языка C++.

1. Базовые типы данных в языке C++.
2. Модификаторы типа и квалификаторы const, volatile, mutable.

3. Производные типы данных: указатели, ссылки, массивы.
4. Классы и объекты в C++: концепция инкапсуляции и наследования.
5. Шаблоны классов и обобщённое программирование.
6. Перечисления (enums) и перечисляемые классы (enum class).
7. Управление памятью и умные указатели (smart pointers).
8. Полиморфизм и переопределение виртуальных функций.
9. Абстрактные базовые классы и интерфейсы в C++.
10. Исключительные ситуации и обработка исключений.
11. Конструкторы и деструкторы объектов в C++.
12. Константные выражения и constexpr-функции.
13. Пространства имен (namespaces) и их влияние на структуру проекта.
14. STL-контейнеры и итерации по контейнерам.
15. Стандартные библиотеки и их роль в расширении возможностей языка C++.

Тема 7. Целочисленные типы. Символьный тип.

1. Представление целых чисел в двоичной форме и диапазон возможных значений.
2. Типы signed int и unsigned int: различия и ограничения.
3. Размер целого числа и разрядность в различных языках программирования.
4. Работа с беззнаковыми (unsigned) целыми числами и битовые операции.
5. Операции над целыми числами: арифметические, логические и побитовые операторы.
6. Преобразование типов и потенциальные ловушки при смешивании знаковых и беззнаковых типов.
7. Специальные целочисленные типы в языке C/C++: short, long, char.
8. Расширенные целые типы в современном стандарте C++ (uint8_t, uint16_t и др.).
9. Отрицательные значения и представление отрицательных чисел в дополнительном коде.
10. Тип символа char и таблица ASCII символов.
11. Кодировки символов UTF-8, UTF-16 и Unicode.
12. Диапазоны символьных типов и их интерпретация в разных операционных системах.
13. Манипуляция строками и преобразования символов в строки и обратно.
14. Константы и литералы целочисленного и символьного типов.
15. Обобщённые символьные типы wchar_t и их применение в мультязычной среде.

Тема 8. Булевский (логический) тип. Вещественные типы.

1. Логический тип bool и булевская алгебра в программировании.
2. Возможные значения логического типа и правила преобразований.
3. Булевские операции AND, OR, NOT и XOR в различных языках программирования.

4. Истинность и ложность в языках программирования высокого уровня.
5. Булевские выражения и оператор ветвления if-else.
6. Применение логических типов в проверке условий и фильтрации данных.
7. Флаги и маски состояний в низкоуровневом программировании.
8. Логические конструкции и управление потоком выполнения программы.
9. Исторически первые логические типы и их развитие в современных языках.
10. Практическое использование логических типов в структурах данных.
11. Основные концепции вещественных типов: float и double.
12. Внутреннее представление вещественных чисел в IEEE 754 стандарт.
13. Потеря точности при операциях с вещественными числами.
14. Округление и точность вычислений с вещественными значениями.
15. Оптимизация расчетов с вещественными числами и предотвращение погрешностей.

Тема 9. Описание переменных и констант. Арифметические операции.

1. Переменные: определение, объявление и инициализация.
2. Основные типы переменных и их диапазоны значений.
3. Правила наименования переменных в различных языках программирования.
4. Постоянные величины (константы) и их использование в программах.
5. Ключевые слова и спецификаторы для объявления констант (const, final, readonly).
6. Границы областей видимости переменных и механизмов их хранения.
7. Приведение типов данных и проблемы приведения типов в арифметических операциях.
8. Важнейшие арифметические операции: сложение, вычитание, умножение, деление.
9. Остаточные операции (%): особенность и сфера применения.
10. Инкремент и декремент: префиксная и постфиксная формы записи.
11. Побитовые арифметические операции и их практическое применение.
12. Приоритеты и ассоциативность арифметических операций.
13. Проблема переполнения и потери точности при выполнении арифметических операций.
14. Поддержка комплексных и рациональных чисел в некоторых языках программирования.
15. Особенности целочисленной и вещественной арифметики в различных средах программирования.

Тема 10. Поразрядные операции языка C++. Операторы языка.

1. Введение в поразрядные операции и их применение в программировании.
2. Битовые операторы: сдвиг вправо и сдвиг влево.
3. Поразрядные логические операции: AND (&), OR (|), XOR (^).

4. Оператор дополнения (~) и инвертирование битов.
5. Таблица истинности для поразрядных операций.
6. Преобразования данных с использованием битовых масок.
7. Порядок выполнения операций и приоритет операторов в C++.
8. Основные группы операторов: арифметические, присваивания, сравнения.
9. Составные операторы присваивания: +=, -=, *=, /=, %=.
10. Тернарный оператор ?: и его особенности.
11. Адресные и указательные операторы (*, &).
12. Индексация массива и доступ к элементам с помощью оператора [].
13. Последовательность выполнения выражений и побочные эффекты операторов.
14. Сложносоставные операторы и группировка выражений с круглыми скобками.
15. Оптимизация программ с применением поразрядных операций и эффективных конструкций.

Тема 11. Средства ввода-вывода. Условный оператор.

1. Ввод и вывод данных в консольных приложениях на языке C++.
2. Библиотека <iostream> и потоки cin/cout.
3. Форматированный ввод-вывод с использованием манипуляторов (setw(), setprecision() и др.).
4. Объекты файла: чтение и запись с помощью fstream, ofstream, ifstream.
5. Организация диалогов с пользователем средствами стандартного ввода-вывода.
6. Консольный интерфейс: получение данных с клавиатуры и отображение результатов.
7. Функция scanf() и printf(): достоинства и недостатки в сравнении с std::cin/std::cout.
8. Работа с файлами бинарных данных в C++.
9. Потоки ввода-вывода и буферизация данных.
10. Условный оператор if: логика ветвления и альтернативные варианты выполнения.
11. Многострочный условный оператор else if и каскады проверок.
12. Условный оператор switch-case и конструкция break.
13. Альтернативные подходы к управлению потоком выполнения программы.
14. Комбинация условных операторов с циклами for, while, do-while.
15. Логические операторы &&, ||, ! и приоритеты их выполнения в условиях.

Тема 12. Оператор выбора. Операторы цикла.

1. Оператор множественного выбора switch: основы и примеры использования.
2. Конструкция case/default и механизмы выхода из оператора switch.

3. Переход к следующему варианту case посредством ключевого слова `fallthrough`.
4. Недостатки оператора `switch` и альтернатива с цепочками условных операторов `if-else`.
5. Оптимизация и производительность переключателей в компилируемых языках.
6. Основные разновидности циклов: цикл с условием (`for`), цикл с проверкой условия в конце (`do-while`) и цикл с предварительным условием (`while`).
7. Управляющие операторы в циклах: `break`, `continue`, `goto`.
8. Итерационные конструкции с поддержкой инкремента и декремента (оператор `++` и `--`).
9. Бесшовное вложение циклов: матричная обработка данных и вложенные структуры.
10. Сравнение циклов `for-each` и традиционных циклов в языках высокого уровня.
11. Безусловные переходы и опасность злоупотребления оператором `goto`.
12. Техника остановки бесконечного цикла и создание безопасной логики обхода ситуаций `deadlock`.
13. Особые случаи применения операторов цикла в конкретных задачах (например, перебор элементов коллекции).
14. Использование комбинированных операторов цикла для обработки иерархических данных.
15. Подходы к выбору оптимального вида цикла в зависимости от конкретной задачи и используемого языка программирования.

Тема 13. Операторы перехода. Обработка последовательностей.

1. Операторы безусловного перехода (`goto`, `break`, `continue`).
2. Логика работы операторов прерывания цикла (`break`) и пропуска итерации (`continue`).
3. Практическое применение оператора безусловного перехода `goto` и риски его использования.
4. Предупреждения компилятора и соглашения стиля кода по применению операторов перехода.
5. Организация потока выполнения программы с использованием операторов `return` и `exit`.
6. Оптимизация последовательностей операторов перехода для упрощения понимания и поддержки кода.
7. Стратегии эффективной обработки линейных последовательностей данных (массивов, списков).
8. Структурная обработка одномерных и двумерных последовательностей.
9. Обработка специфичных случаев последовательности: пустые элементы, уникальные значения.
10. Паттерны обработки последовательностей с применением стандартных библиотек (STL).

11. Обработка ошибок и некорректных данных в рамках последовательностей.
12. Генерация и модификация последовательностей с помощью стандартных библиотечных функций.
13. Изучение способов повышения производительности при обработке длинных последовательностей.
14. Особенности обработки последовательностей разнотипных данных.
15. Оптимизация алгоритмов обработки последовательностей для снижения потребления памяти и ускорения выполнения.

Тема 14. Итерационные циклы. Программирование вложенных. Циклов.

1. Понятие итерационного цикла и его применение в программировании.
2. Основные типы итерационных циклов: for, while, do-while.
3. Формы записи цикла for и особенности их использования.
4. Постусловный цикл do-while и сценарии его применения.
5. Инварианты цикла и контроль корректности вывода.
6. Оптимизация итерационных циклов для уменьшения временных затрат.
7. Организация вложенных циклов и способы предотвращения «проблемы квадратичного роста».
8. Постановка границ вложенности и профилактика зависших процессов.
9. Вложенный цикл как инструмент для работы с многомерными массивами.
10. Многопоточная обработка вложенных циклов: синхронизация и безопасность.
11. Генеративные циклы и создание уникальных комбинаций данных.
12. Автоматизация настройки количества уровней вложенности.
13. Решение задачи обхода дерева или графа с помощью вложенных циклов.
14. Особенности программной реализации вложенных циклов в различных парадигмах программирования.
15. Профилирование и тестирование производительности вложенных циклов.

Тема 15. Регулярные типы. Обработка одномерных массивов.

1. Понятие регулярного типа и его значимость в программировании.
2. Одномерные массивы: общие характеристики и способы объявления.
3. Процедуры заполнения и чтения одномерных массивов.
4. Поиск максимальных и минимальных элементов в массивах.
5. Сортировка одномерных массивов: пузырьковый метод, быстрая сортировка.
6. Линейный поиск элемента в массиве и его модификации.
7. Частотные таблицы и статистика распределений элементов массива.
8. Изменение размера одномерного массива и динамические массивы.
9. Удаление и вставка элементов в одномерный массив.
10. Прямой проход по массиву и обратная итерация.

11. Суммирование и усреднение элементов массива.
12. Двухпроходные алгоритмы обработки одномерных массивов.
13. Преобразование массивов и копирование данных.
14. Анализ пропусков нулевых элементов и подсчет ненулевых значений.
15. Оптимизация производительности обработки массивов в различных языках программирования.

Тема 16. Классы задач по обработке массивов. Задачи 1-ого и 2-го класса.

1. Классификация задач по степени сложности обработки массивов.
2. Первоклассные задачи (1 класс): элементарные операции над элементами массива.
3. Второклассные задачи (2 класс): упорядоченность и поиск закономерностей.
4. Оценка трудоемкости решения задач различных классов.
5. Алгоритмы поиска элементов в отсортированном массиве.
6. Применение рекурсии для решения задач второго класса.
7. Алгоритмы обработки массивов в параллельном режиме.
8. Сравнительная характеристика задач первого и второго класса.
9. Критерии отнесения задачи к первому или второму классу.
10. Оптимизация решения задач первого класса для увеличения производительности.
11. Связь между уровнем задачи и выбором подходящего метода обработки.
12. Задача нахождения среднего значения и минимальная сумма абсолютных отклонений.
13. Вычисление медианы массива и построение гистограммы распределения.
14. Задача выделения экстремумов и построения огибающей кривой.
15. Эффективные подходы к решению задач обоих классов с учётом ограничений памяти и скорости вычислений.

Тема 17. Классы задач по обработке массивов. Задачи 3-ого и 4-го класса.

1. Третий класс задач: задачи повышенной сложности и нестандартные алгоритмы.
2. Четвертый класс задач: высокая степень абстракции и сложные математические модели.
3. Комплексные процедуры поиска и выборки элементов в массивах третьего класса.
4. Алгоритмы поиска кратчайшего пути и оптимальное покрытие множества четвертого класса.
5. Матрицы смежности и смежные задачи четвёртого класса.
6. Интерактивные задачи и обучение в третьем классе.
7. Парные перестановки и алгоритмы транспонирования в четвёртом классе.

8. Упорядоченное хранение и быстрое извлечение данных в задачах третьего класса.
9. Смешанные алгоритмы и гибридные методики обработки массивов четвертого класса.
10. Решение задач третьего класса с привлечением теории графов.
11. Направленный поиск локального экстремума и формирование подмножеств в задачах четвертого класса.
12. Проверка принадлежности объекта группе и выделение подгрупп.
13. Оптимизация сложных запросов к массивам третьего и четвертого класса.
14. Поиск максимальной общей подпоследовательности в задачах четвертого класса.
15. Классические и новые подходы к эффективному решению задач высших классов.

Тема 18. Двумерные массивы.

1. Определение и общая структура двумерных массивов.
2. Объявление и инициализация двумерных массивов.
3. Доступ к элементам двумерного массива и индексация.
4. Базовые операции с двумерными массивами: заполнение и считывание.
5. Транспонирование матриц и зеркальное отражение двумерных массивов.
6. Обработка строк и столбцов в двумерных массивах.
7. Нахождение суммы элементов вдоль диагоналей матрицы.
8. Обнаружение и удаление дубликатов в двумерных массивах.
9. Сортировка двумерных массивов по строкам и столбцам.
10. Алгоритмы поиска элементов в частично упорядоченном двумерном массиве.
11. Решение уравнений методом Гаусса с использованием двумерных массивов.
12. Умножение матриц и векторные произведения.
13. Эффективные алгоритмы вращения двумерных массивов.
14. Продвинутое техники поиска максимумов и минимумов в двумерных массивах.
15. Оптимизация обработки двумерных массивов для больших объемов данных.

Тема 19. Функции. Прототип функции. Параметры функции.

1. Определение и назначение функций в программировании.
2. Создание прототипа функции и его важность в разработке.
3. Объявление и реализация функций в различных языках программирования.
4. Разновидности параметров функций: обязательные и необязательные.
5. Значения по умолчанию для параметров функций.
6. Перегрузка функций и полиморфизм.

7. Разделение ответственности между функциями и модулями программы.
8. Возвращаемые значения функций и управление состоянием программы.
9. Рекурсивные функции и решение задач с их помощью.
10. Использование анонимных функций и лямбда-выражений.
11. Замыкания и сохранение контекста функции.
12. Трансляция и исполнение функций: интерпретатор vs компилятор.
13. Оптимизация вызовов функций и снижение накладных расходов.
14. Аргументы функции и эффект побочных эффектов.
15. Передача параметров по ссылке и по значению: отличия и последствия.

Тема 20. Локальные и глобальные переменные. Область видимости переменных.

1. Понятие локальных и глобальных переменных.
2. Правила объявления и использования локальных переменных.
3. Области видимости переменных и уровни доступности.
4. Жизненный цикл локальных и глобальных переменных.
5. Воздействие области видимости на производительность программы.
6. Проблемы конфликта имен и разрешение конфликтов.
7. Закрывание области видимости: блоки и функции.
8. Глобальные переменные и возможные негативные последствия их использования.
9. Правильное использование статических переменных в функциях.
10. Передача данных между областями видимости.
11. Обработка локальных и глобальных переменных в многопоточных приложениях.
12. Обеспечение модульности кода с помощью локальных переменных.
13. Вопросы безопасности и защиты данных при обращении к глобальным переменным.
14. Устойчивость программ к ошибкам благодаря контролю области видимости.
15. Оптимизация производительности за счёт правильного использования областей видимости переменных.

Тема 21. Рекурсия. Шаблоны функций.

1. Понятие рекурсии и ее роль в программировании.
2. Примеры классической рекурсии: факториалы, фибоначчьева последовательность.
3. Глубина рекурсии и пределы стека.
4. Рекурсивные функции и их эффективное использование.
5. Хвостовая рекурсия и ее оптимизация компиляторами.
6. Математические основания рекурсивных вычислений.
7. Ограничения и опасности рекурсии: возврат стека и рекурсивные атаки.
8. Сравнение рекурсивных и итеративных решений.
9. Рекурсивные шаблоны и мета-программирование.
10. Повторное использование шаблонных функций в C++.

- 11.Макросы и шаблоны функций: сходства и различия.
- 12.Обобщённая функциональность и полиморфизм через шаблоны функций.
- 13.Проектирование шаблонных функций для лучшей читаемости и производительности.
- 14.Совместимость шаблонов функций с наследованием и перегрузкой.
- 15.Шаблонные специализации и конкретные реализации функций-шаблонов.

Тема 22. Методы сортировки.

1. Классификация методов сортировки по различным признакам.
2. Сортировка пузырьком (Bubble Sort): базовый алгоритм и его улучшение.
3. Быстрая сортировка (QuickSort): теория и практика применения.
4. Сортировка вставками (Insertion Sort): область применения и оптимизация.
5. Сортировка выбором (Selection Sort): простой и эффективный подход.
6. Сортировка слиянием (Merge Sort): принципы работы и реализация.
7. Радиционная сортировка (Radix Sort): специальные случаи и требования к входным данным.
8. Медианная сортировка (Median Sort): особые случаи и адаптивность.
9. Распределительная сортировка (Bucket Sort): распределение элементов по корзинам.
10. Турнирная сортировка (Tournament Sort): построение турнирных деревьев.
11. Гибридные методы сортировки: сочетание двух и более алгоритмов.
12. Внешняя сортировка: сортировка данных большого объема.
13. Непростая задача быстрой сортировки больших объёмов данных.
14. Выбор оптимального метода сортировки в зависимости от особенностей данных.
15. Современное состояние исследований и разработка новых методов сортировки.

Тема 23. Указатели. Динамические переменные.

1. Понятие указателя и адресная арифметика.
2. Работа с указателями и массивами в языках программирования.
3. Обращение к динамическим переменным через указатели.
4. Выделения и освобождение памяти в динамических структурах.
5. Проблемы утечек памяти и неправильное освобождение ресурсов.
6. Нулевые указатели и их правильное использование.
7. Двойные указатели и косвенная адресация.
8. Использование указателей на функции и передачу их в качестве аргументов.
9. Указатели на члены классов и структуры данных.
10. Массивы указателей и динамически выделенные массивы.

11. Указатели и система управления памятью в операционной системе.
12. Современные подходы к безопасным манипуляциям с указателями.
13. Библиотеки и фреймворки для удобной работы с динамическими объектами.
14. Применения указателей в высокоэффективных библиотеках и API-интерфейсах.
15. Оптимизация работы с указателями и повышение производительности программ.

Тема 24. Динамические массивы.

1. Определение и необходимость динамических массивов.
2. Сравнение статических и динамических массивов.
3. Основные операции с динамическими массивами: создание, изменение размеров, удаление.
4. Реализация динамических массивов с помощью указателей.
5. Проблемы фрагментации памяти при изменении размера динамических массивов.
6. Автоматическое расширение динамических массивов и стратегия удвоения.
7. Эффективные алгоритмы поиска и сортировки в динамических массивах.
8. Работа с векторами и аналогичными структурами в стандартной библиотеке C++.
9. Использование динамических массивов в объектах классов и конструкторах.
10. Повышение производительности динамических массивов за счет предварительной резервации памяти.
11. Хранение и манипулирование большими объемами данных с помощью динамических массивов.
12. Алгоритмы сжатия и уплотнения динамических массивов.
13. Построение многомерных динамических массивов и их применение.
14. Оптимизация работы с индексами и ускорение доступа к элементам динамических массивов.
15. Организация автоматической очистки памяти при завершении работы с динамическими массивами.

Критерии оценки выполнения рефератов по учебной дисциплине

Оценка «отлично» выставляется студенту, если тема реферата раскрыта в полной мере и все требования по написанию реферата соблюдены.

Оценка «хорошо» выставляется студенту, если недостаточно раскрыта тема реферата, но все требования по написанию реферата соблюдены.

Оценка «удовлетворительно» выставляется студенту, если недостаточно раскрыта тема реферата, не все требования по написанию реферата соблюдены, присутствуют ошибки и неточности в работе.

Оценка «неудовлетворительно» выставляется студенту, если отсутствует реферат.

Рекомендации по написанию реферата:

Реферат — письменная работа, выполняемая обучающимся в течение определенного срока.

Реферат — краткое точное изложение сущности какого-либо вопроса, темы на основе одной или нескольких книг, монографий или других первоисточников. Реферат должен содержать основные фактические сведения и выводы по рассматриваемому вопросу.

Реферат отвечает на вопрос — что содержится в данной публикации (публикациях). Реферат — не механический пересказ работы, а изложение ее сущности.

Кроме реферирования прочитанной литературы, от обучающегося требуется аргументированное изложение собственных мыслей по рассматриваемому вопросу. Тему реферата предлагает преподаватель или сам обучающийся, в последнем случае она должна быть согласована с преподавателем.

В реферате нужны развернутые аргументы, рассуждения, сравнения. Материал подается не столько в развитии, сколько в форме констатации или описания.

Содержание реферируемого произведения излагается объективно от имени автора. Если в первичном документе главная мысль сформулирована недостаточно четко, в реферате она должна быть конкретизирована и выделена.

Структура реферата:

1. Титульный лист.
2. После титульного листа на отдельной странице следует оглавление (план, содержание), в котором указаны названия всех разделов (пунктов плана) реферата и номера страниц, указывающие начало этих разделов в тексте реферата.
3. После оглавления следует введение. Объем введения составляет 1,5-2 страницы.
4. Основная часть реферата может иметь одну или несколько глав, состоящих из 2-3 параграфов (подпунктов, разделов) и предполагает осмысленное и логичное изложение главных положений и идей, содержащихся в изученной литературе. В тексте обязательны ссылки на первоисточники. В том случае если цитируется или используется чья-либо неординарная мысль, идея, вывод, приводится какой-либо цифрой материал, таблицу - обязательно сделайте ссылку на того автора у кого вы взяли данный материал.
5. Заключение содержит главные выводы, и итоги из текста основной части, в нем отмечается, как выполнены задачи и достигнуты ли цели, сформулированные во введении.

6. Приложение может включать графики, таблицы, расчеты.

7. Библиография (список литературы) здесь указывается реально использованная для написания реферата литература. Список составляется согласно правилам библиографического описания.

Этапы работы над рефератом.

Работу над рефератом можно условно подразделить на три этапа:

1. Подготовительный этап, включающий изучение предмета исследования;

2. Изложение результатов изучения в виде связного текста;

3. Устное сообщение по теме реферата.

1. Подготовительный этап работы.

Формулировка темы. Подготовительная работа над рефератом начинается с формулировки темы. Тема в концентрированном виде выражает содержание будущего текста, фиксируя как предмет исследования, так и его ожидаемый результат. Для того чтобы работа над рефератом была успешной, необходимо, чтобы тема заключала в себе проблему, скрытый вопрос (даже если наука уже давно дала ответ на этот вопрос, студент, только знакомящийся с соответствующей областью знаний, будет вынужден искать ответ заново, что даст толчок к развитию проблемного, исследовательского мышления).

Поиск источников. Грамотно сформулированная тема зафиксировала предмет изучения; задача обучающегося — найти информацию, относящуюся к данному предмету и разрешить поставленную проблему. Выполнение этой задачи начинается с поиска источников. На этом этапе необходимо вспомнить, как работать с энциклопедиями и энциклопедическими словарями.

Работа с источниками. Работу с источниками надо начинать с ознакомительного чтения, т. е. просмотреть текст, выделяя его структурные единицы. При ознакомительном чтении закладками отмечаются те страницы, которые требуют более внимательного изучения. В зависимости от результатов ознакомительного чтения выбирается дальнейший способ работы с источником. Если для разрешения поставленной задачи требуется изучение некоторых фрагментов текста, то используется метод выборочного чтения. Если в книге нет подробного оглавления, следует обратить внимание на предметные и именные указатели.

Избранные фрагменты или весь текст (если он целиком имеет отношение к теме) требуют вдумчивого, неторопливого чтения с «мысленной проработкой» материала. Такое чтение предполагает выделение: 1) главного в тексте; 2) основных аргументов; 3) выводов. Особое внимание следует обратить на то, вытекает тезис из аргументов или нет. Необходимо также проанализировать, какие из утверждений автора носят проблематичный, гипотетический характер и уловить скрытые вопросы.

Понятно, что умение таким образом работать с текстом приходит далеко не сразу. Наилучший способ научиться выделять главное в тексте, улавливать проблематичный характер утверждений, давать оценку авторской позиции — это сравнительное чтение, в ходе которого студент знакомится с различными

мнениями по одному и тому же вопросу, сравнивает весомость и доказательность аргументов сторон и делает вывод о наибольшей убедительности той или иной позиции.

Создание конспектов для написания реферата.

Подготовительный этап работы завершается созданием конспектов, фиксирующих основные тезисы и аргументы. Здесь важно вспомнить, что конспекты пишутся на одной стороне листа, с полями и достаточным для исправления и ремарок межстрочным расстоянием (эти правила соблюдаются для удобства редактирования). Если в конспектах приводятся цитаты, то непременно должно быть дано указание на источник (автор, название, выходные данные, № страницы).

По завершении предварительного этапа можно переходить непосредственно к созданию текста реферата.

2. Создание текста.

Общие требования к тексту.

Текст реферата должен подчиняться определенным требованиям: он должен раскрывать тему, обладать связностью и цельностью.

Раскрытие темы предполагает, что в тексте реферата излагается относящийся к теме материал и предлагаются пути решения содержащейся в теме проблемы; связность текста предполагает смысловую соотносительность отдельных компонентов, а цельность - смысловую законченность текста.

С точки зрения связности все тексты делятся на тексты-констатации и тексты-рассуждения. Тексты-констатации содержат результаты ознакомления с предметом и фиксируют устойчивые и несомненные суждения. В текстах-рассуждениях одни мысли извлекаются из других, некоторые ставятся под сомнение, дается им оценка, выдвигаются различные предположения.

План реферата.

Универсальный план реферата – введение, основной текст и заключение.

Требования к введению.

Во введении аргументируется актуальность исследования, - т. е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками; перечисляются положения, которые должны быть обоснованы. Введение может также содержать обзор источников или экспериментальных данных, уточнение исходных понятий и терминов, сведения о методах исследования. Во введении обязательно формулируются цель и задачи реферата.

Объем введения - в среднем около 10% от общего объема реферата.

Основная часть реферата.

Основная часть реферата раскрывает содержание темы. Она наиболее значительна по объему, наиболее значима и ответственна. В ней обосновываются основные тезисы реферата, приводятся развернутые аргументы, предполагаются гипотезы, касающиеся существа обсуждаемого вопроса. Важно проследить, чтобы основная часть не имела форму монолога. Аргументируя собственную позицию, можно и должно анализировать, и оценивать позиции различных исследователей, с чем-то соглашаться, чему-то

возражать, кого-то опровергать. Текст основной части делится на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки материала: классификации (эмпирические исследования), типологии (теоретические исследования), периодизации (исторические исследования).

Заключение.

Заключение — последняя часть научного текста. В ней краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования. Здесь же могут намечаться и дальнейшие перспективы развития темы. Небольшое по объему сообщение также не может обойтись без заключительной части - пусть это будут две-три фразы. Но в них должен подводиться итог проделанной работы.

Список использованной литературы.

Реферат любого уровня сложности обязательно сопровождается списком используемой литературы. Названия книг в списке располагают по алфавиту с указанием выходных данных использованных книг.

Требования, предъявляемые к оформлению реферата

Объемы рефератов – от 10-18 машинописных страниц. Работа выполняется на одной стороне листа стандартного формата. По обеим сторонам листа оставляются поля размером 35 мм. слева и 15 мм. справа, рекомендуется шрифт 12-14, интервал - 1,5. Все листы реферата должны быть пронумерованы. Каждый вопрос в тексте должен иметь заголовок в точном соответствии с наименованием в плане-оглавлении. При написании и оформлении реферата следует избегать типичных ошибок, например, таких:

- поверхностное изложение основных теоретических вопросов выбранной темы, когда автор не понимает, какие проблемы в тексте являются главными, а какие второстепенными,

- в некоторых случаях проблемы, рассматриваемые в разделах, не раскрывают основных аспектов выбранной для реферата темы,

- дословное переписывание книг, статей, заимствования рефератов из интернета и т. д.

При проверке реферата преподавателем оцениваются:

1. Знания и умения на уровне требований стандарта конкретной дисциплины: знание фактического материала, усвоение общих представлений, понятий, идей.

2. Характеристика реализации цели и задач исследования (новизна и актуальность поставленных в реферате проблем, правильность формулирования цели, определения задач исследования, правильность выбора методов решения задач и реализации цели; соответствие выводов решаемым задачам, поставленной цели, убедительность выводов).

3. Степень обоснованности аргументов и обобщений (полнота, глубина, всесторонность раскрытия темы, логичность и последовательность изложения материала, корректность аргументации и системы доказательств, характер и достоверность примеров, иллюстративного материала, широта кругозора автора, наличие знаний интегрированного характера, способность к

обобщению).

4. Качество и ценность полученных результатов (степень завершенности реферативного исследования, спорность или однозначность выводов).

5. Использование литературных источников.

6. Культура письменного изложения материала.

7. Культура оформления материалов работы.

Комплект типовых практических заданий

Тема 1. Алгоритм и его свойства

Определение алгоритма и его ключевых характеристик.

Основные этапы формирования и реализации алгоритма.

Классификация алгоритмов по назначению и способу реализации.

Основные свойства алгоритмов: конечность, массовость, детерминированность, результативность.

Примеры простых и сложных алгоритмов.

Вопросы для самоконтроля:

1. Что такое алгоритм и какими свойствами он должен обладать?
2. В чём разница между процедурными и декларативными языками программирования?
3. Почему важна детализация этапов при создании алгоритма?
4. Для чего нужен псевдокод при разработке алгоритмов?
5. Что означает термин «детерминированность» применительно к алгоритмам?

Цель работы:

Формирование представлений об основах построения и реализации алгоритмов, понимание их свойств и важности строгой постановки задач.

В работе используется презентация слайдов выполненная в MS Powerpoint

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 2. Понятие сегмента и регистра. Структура ЦП.

Определение и назначение регистров и сегментов в структуре центрального процессора (ЦП).

Классификация регистров: регистры данных, управляющие регистры, сегментные регистры.

Структура современного центрального процессора: основные узлы и взаимосвязи.

Режимы работы процессора: реальный режим, защищённый режим, режим Long Mode.

Понятие адресации памяти и роль сегментных регистров в этой процедуре.

Вопросы для самоконтроля:

1. Что такое сегмент памяти и зачем он нужен?
2. Какие существуют основные типы регистров в центральном процессоре?
3. Какова роль сегментных регистров в механизме виртуальной памяти?
4. В чём состоит различие реального режима и защищённого режима работы процессора?
5. Как организуется обращение к ячейкам памяти с использованием сегментных регистров?

Цель работы:

Изучить основные элементы архитектуры центрального процессора, освоить классификацию регистров и разобраться в назначении сегментных регистров.

В работе используется презентация слайдов выполненная в MS Powerpoint

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 3. Функции языков программирования.

Определение функций в языках программирования и их роль.

Классификация функций: встроенные, стандартные, пользовательские.

Форматы объявлений и вызовов функций.

Способы передачи параметров в функции: по значению, по ссылке, по указателю.

Примеры реализации функций на разных языках программирования (Python, C++, JavaScript).

Вопросы для самоконтроля:

1. Что такое функция в языках программирования и какую пользу она приносит разработчику?
2. Какие бывают категории функций и в чём заключаются их особенности?
3. В чём разница между передачей параметра по значению и по ссылке?
4. Как определить возвращаемое значение функции и зачем оно нужно?
5. Каково основное преимущество использования собственных функций в крупных проектах?

Цель работы:

Освоить понятие функций в языках программирования, научиться объявлять собственные функции и понимать разницу между разными способами передачи параметров.

В работе используется база данных выполненная в MS Access/

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 4. Метаязыки описания языков программирования

Определение метаязыков и их назначение.

Виды метаязыков: лексико-грамматические, формальные грамматики, языки разметки.

Методы описания синтаксиса и семантики языков программирования.

Пример метаязыка BNF (Backus-Naur Form) и его применение.

Современные инструменты для генерации парсеров и анализаторов.

Вопросы для самоконтроля:

1. Что такое метаязык и почему он необходим при описании языков программирования?
2. Какие существуют виды метаязыков и в чём состоят их различия?
3. Как устроены формальные грамматики и как они применяются для описания синтаксиса?
4. Что представляет собой форма Бэкуса-Наура (BNF)? В каких случаях она применяется?
5. Зачем нужны инструменты автоматического синтеза парсеров и какие из них наиболее популярны?

Цель работы:

Получить знания о роли метаязыков в описании и формальном представлении языков программирования, освоив на практике форму Бэкуса-Наура и современные инструменты для синтаксического анализа.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может

исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 5. Виртуальная машина.

Определение виртуальной машины и её роль в компьютерных системах.

Типы виртуальных машин: системные, процессы, специализированные.

Основные компоненты виртуальной машины: процессор, память, устройства ввода-вывода.

Примеры известных виртуальных машин: JVM (Java Virtual Machine), CLR (.NET Common Language Runtime), PyPy (для Python).

Преимущества и недостатки использования виртуальных машин.

Вопросы для самоконтроля:

1. Что такое виртуальная машина и какие задачи она решает?
2. Какие существуют типы виртуальных машин и в чём их принципиальные различия?
3. Как устроены основные компоненты виртуальной машины?
4. Как работают известные виртуальные машины вроде JVM и CLR?
5. В чём плюсы и минусы использования виртуальных машин в разработке программного обеспечения?

Цель работы:

Познакомиться с концепцией виртуальных машин, их видами и компонентами, а также понять роль виртуальных машин в современном программировании.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 6. Структура типов языка C++

Определение типов данных и их роль в языке C++.

Классификация типов данных: примитивные, производные, определяемые пользователем.

Примитивные типы данных: целочисленные, вещественные, логические, символьные.

Производные типы данных: массивы, указатели, ссылки, объединения, структуры.

Пользовательские типы данных: классы и конструкторы.

Вопросы для самоконтроля:

1. Что такое тип данных и зачем они используются в языке C++?
2. Какие бывают виды примитивных типов данных и в чём их особенности?
3. Что представляют собой производные типы данных и как они создаются?
4. Как реализуются пользовательские типы данных в C++?
5. В чём разница между указателями и ссылками в языке C++?

Цель работы:

Ознакомиться с основными типами данных языка C++, особенностями их объявления и использования, сформировать навыки грамотного выбора типов данных для решения прикладных задач.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью..

Тема 7. Целочисленные типы. Символьный тип

Определение целочисленных типов и их разновидностей в языке программирования.

Интервал значений и размеры целочисленных типов: int, short, long, byte.

Знаковые и беззнаковые целочисленные типы.

Символьный тип `char` и его особенности.

Таблица ASCII и юникодные символы.

Вопросы для самоконтроля:

1. Что такое целочисленные типы данных и какие они бывают?
2. Чем отличаются знаковые и беззнаковые целочисленные типы?
3. Какие размерности имеют целочисленные типы в языке программирования?
4. Что представляет собой символьный тип данных `char`?
5. Как связаны коды символов в таблице ASCII и символы Юникода?

Цель работы:

Изучить целочисленные и символьные типы данных, ознакомиться с их характеристиками и правилами использования в программировании.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 8. Булевский (логический) тип. Вещественные типы.

Определение булевского (логического) типа и его роль в языках программирования.

Логические операции и их обозначение: AND, OR, NOT, XOR.

Применение булевского типа в условиях и логических выражениях.

Определение вещественных типов и их использование в языках программирования.

Различные форматы представления вещественных чисел: `float`, `double`, `decimal`.

Вопросы для самоконтроля:

1. Что такое булевский тип данных и как он используется в программах?

2. Какие логические операции поддерживаются языком программирования?
3. В чём разница между типами float и double?
4. Как вещественные типы влияют на точность вычислений?
5. Где чаще всего используются вещественные типы данных?

Цель работы:

Научиться применять булевский и вещественный типы данных в языке программирования, осознать их различия и особенности использования.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 9. Описание переменных и констант. Арифметические операции.

Определение переменных и констант в языках программирования.

Объявление и инициализация переменных различного типа.

Правила именования переменных и рекомендации по стилю оформления.

Основные арифметические операции (+, -, *, /, %).

Примеры работы с арифметическими операциями в программах.

Вопросы для самоконтроля:

1. Что такое переменная и как правильно её объявить?
2. Чем отличается переменная от константы?
3. Какие правила действуют при выборе имени переменной?
4. Какие арифметические операции поддерживает большинство языков программирования?
5. Как работает операция деления нацело (%) и в каких ситуациях она полезна?

Цель работы:

Освоить технику объявления переменных и констант, изучить работу основных арифметических операций и закрепить полученные знания на практике.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 10. Поразрядные операции языка C++. Операторы языка.

Понятие поразрядных операций и их роль в языке C++.

Основные поразрядные операторы: `&`, `|`, `^`, `~`, `<<`, `>>`.

Применение поразрядных операций для манипуляции отдельными битами.

Классификация операторов языка C++: арифметические, логические, отношения, присвоения.

Приоритет и ассоциативность операторов в выражениях.

Вопросы для самоконтроля:

1. Что такое поразрядные операции и какие они бывают?
2. Какими операторами выполняются поразрядные операции в C++?
3. Как отличить операцию логического AND от поразрядного AND?
4. Какие виды операторов имеются в языке C++?
5. Влияют ли приоритеты операторов на порядок вычисления выражений?

Цель работы:

Изучить основы поразрядных операций и освоить использование различных операторов языка C++, включая их приоритет и ассоциативность.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 11. Средства ввода-вывода. Условный оператор.

Основные функции и методы ввода-вывода в языках программирования.

Стандартные потоки ввода-вывода: `stdin`, `stdout`, `stderr`.

Работа с файловым вводом-выводом: открытие, закрытие, чтение и запись.

Понятие условного оператора и его роль в принятии решений программой.

Примеры использования условных операторов (`if`, `switch-case`) в программировании.

Вопросы для самоконтроля:

1. Какие существуют стандартные потоки ввода-вывода и как ими пользоваться?
2. Как читать и записывать данные в файлы?
3. Что такое условный оператор и какие формы он принимает?
4. В чём разница между операторами `if` и `switch-case`?
5. Как избежать распространённых ошибок при использовании условных операторов?

Цель работы:

Изучить основные средства ввода-вывода и научиться эффективно использовать условные операторы для управления ходом выполнения программы.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и

аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 12. Оператор выбора. Операторы цикла.

Понятие оператора выбора и его виды (if, switch-case).

Использование оператора if для принятия решений в программах.

Особенности и преимущества оператора switch-case.

Классификация операторов цикла: for, while, do-while.

Примеры применения циклов и оператора выбора в реальной практике программирования.

Вопросы для самоконтроля:

1. Что такое оператор выбора и какие формы он имеет?
2. Чем отличается оператор if от оператора switch-case?
3. Какие операторы цикла предусмотрены в большинстве языков программирования?
4. В каких случаях предпочтительнее использовать цикл for, а в каких — while?
5. Какие особенности важно учитывать при работе с оператором switch-case?

Цель работы:

Изучить операторы выбора и циклы, научиться грамотно выбирать подходящий оператор для конкретной задачи и правильно составлять логику программы.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 13. Операторы перехода. Обработка последовательностей.

Определение операторов перехода и их применение в языках программирования.

Основные операторы перехода: goto, continue, break, return.

Влияние операторов перехода на ход выполнения программы.

Способы обработки последовательностей данных: линейные и циклические алгоритмы.

Примеры использования операторов перехода для обработки последовательностей.

Вопросы для самоконтроля:

1. Что такое операторы перехода и зачем они нужны?
2. Какие операторы перехода существуют в языке программирования?
3. Как влияет оператор break на выполнение цикла?
4. Как обрабатываются большие последовательности данных?
5. Какие ошибки возникают при неправильном применении операторов перехода?

Цель работы:

Изучить операторы перехода и освоить приёмы эффективного использования циклов и линейных алгоритмов для обработки последовательностей данных.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 14. Итерационные циклы. Программирование вложенных Циклов.

Понятие итерационного цикла и его роль в программировании.

Основные виды итерационных циклов: for, while, do-while.

Основные признаки вложенного цикла и примеры их использования.

Взаимодействие внешнего и внутреннего циклов.

Типичные задачи, решаемые с помощью вложенных циклов (обработка таблиц, графика, обработка изображений).

Вопросы для самоконтроля:

1. Что такое итерационный цикл и каким образом он функционирует?
2. Какие существуют основные виды итерационных циклов?
3. В каких случаях удобно использовать вложенные циклы?
4. Как организовать правильную обработку данных с помощью вложенных циклов?
5. Какие проблемы могут возникать при неверной организации вложенных циклов?

Цель работы:

Освоить концепцию итерационных циклов и научиться эффективно решать задачи с использованием вложенных циклов.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 15. Регулярные типы. Обработка одномерных массивов.

Понятие регулярных типов и их роль в программировании.

Объявление и инициализация одномерных массивов.

Индексирование и доступ к элементам одномерного массива.

Основные операции с одномерными массивами: поиск, сортировка, суммирование.

Практические задачи по обработке одномерных массивов (поиск минимума/максимума, фильтрацию, переворот массива).

Вопросы для самоконтроля:

1. Что такое регулярный тип и какие к ним относятся?
2. Как правильно объявить и инициализировать одномерный массив?
3. Какие индексы используются для обращения к элементам массива?
4. Какие операции часто применяют к одномерным массивам?
5. Какие задачи решаются с помощью одномерных массивов?

Цель работы:

Изучить регулярные типы данных, овладеть приёмами работы с одномерными массивами и уметь решать типовые задачи на их основе.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 16. Классы задач по обработке массивов. Задачи 1-ого и 2-го класса.

Определение классов задач по обработке массивов и их классификация.

Примеры задач 1-го класса: поиск, сортировка, минимальные/максимальные элементы.

Примеры задач 2-го класса: частотный анализ, перекomпоновка элементов, накопленная сумма.

Требования и характеристики каждой группы задач.
Методы и алгоритмы решения задач 1-го и 2-го класса.

Вопросы для самоконтроля:

1. Что такое классы задач по обработке массивов?
2. Какие задачи относят к задачам 1-го класса?
3. Какие задачи входят в группу задач 2-го класса?
4. Какие требования предъявляются к задачам разного класса?
5. Какие методы используют для решения задач 1-го и 2-го класса?

Цель работы:

Рассмотреть задачи различной сложности по обработке массивов, научиться отличать задачи разных классов и подбирать подходящие алгоритмы для их решения.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 17. Классы задач по обработке массивов. Задачи 3-ого и 4-го класса.

Определение и особенности задач 3-его и 4-ого класса по обработке массивов.

Примеры задач 3-его класса: перестановки, вращение матрицы, подбор комбинации элементов.

Примеры задач 4-ого класса: задачи оптимальной упаковки, минимальное покрытие, кратчайшие пути.

Методы и алгоритмы решения задач 3-его и 4-ого класса.

Сложности и особенности программирования задач высоких классов.

Вопросы для самоконтроля:

1. Какие задачи относятся к третьему классу?

2. Какие задачи составляют четвертый класс?
3. Какие алгоритмы используются для решения задач третьих и четвертых классов?
4. Какие трудности встречаются при решении задач 3-его и 4-ого класса?
5. Как выбрать оптимальный алгоритм для решения задачи конкретного класса?

Цель работы:

Изучив задачи повышенного уровня сложности (3-его и 4-ого классов), развить навыки программирования и подбора эффективных алгоритмов для их решения.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 18. Двумерные массивы.

Определение двумерных массивов и их назначение.

Объявление и инициализация двумерных массивов.

Работа с элементами двумерного массива: доступ и обновление.

Основные операции с двумерными массивами: поиск, сортировка, поворот, транспонирование.

Примеры задач, решаемых с помощью двумерных массивов (математические задачи, обработка изображений, игры).

Вопросы для самоконтроля:

1. Что такое двумерный массив и для чего он предназначен?
2. Как объявляется и инициализируется двумерный массив?
3. Как обратиться к элементу двумерного массива?
4. Какие операции часто выполняют с двумерными массивами?
5. Какие задачи удобно решать с помощью двумерных массивов?

Цель работы:

Освоить понятие двумерных массивов, приобрести опыт работы с ними и умения решать типовые задачи.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 19. Функции. Прототип функции. Параметры функции.

Понятие функции в программировании и её роль.

Определение и назначение прототипа функции.

Способы передачи параметров в функции: по значению, по ссылке, по указателю.

Виды параметров: обязательные и опциональные.

Вложенные функции и рекурсивные вызовы.

Вопросы для самоконтроля:

1. Что такое функция и для чего она служит в программировании?
2. Что такое прототип функции и зачем он нужен?
3. Какие существуют способы передачи параметров в функцию?
4. Какие бывают виды параметров в функциях?
5. Как работает рекурсия и в каких случаях её целесообразно применять?

Цель работы:

Закрепить знание о функциях, научиться создавать и использовать прототипы функций, освоить разные способы передачи параметров и понять, как применять рекурсию в задачах программирования.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 20. Локальные и глобальные переменные. Область видимости переменных.

Определение локальных и глобальных переменных.

Правила объявления и инициализации переменных.

Область видимости переменных: глобальная, локальная, внутренняя.

Время жизни переменных и проблема «переживающих» переменных.

Способы контроля и управления видимостью переменных.

Вопросы для самоконтроля:

1. Что такое локальные и глобальные переменные?
2. В какой области видимости находится глобальная переменная?
3. Как объявляют локальные переменные и какова их область видимости?
4. Какие проблемы возникают при неправильной работе с видимостями переменных?
5. Как управлять временем жизни переменных?

Цель работы:

Освоить понятия локальных и глобальных переменных, усвоить правила их объявления и области видимости, научиться избегать ошибок, связанных с управлением переменными.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 21. Рекурсия. Шаблоны функций.

Понятие рекурсии и её применение в программировании.

Основные свойства рекурсивных функций: основание и шаг рекурсии.

Примеры рекурсивных алгоритмов: факториал, числа Фибоначчи, Ханойская башня.

Определение шаблона функции и его роль в языке программирования.

Применение шаблонов функций для обобщённого программирования.

Вопросы для самоконтроля:

1. Что такое рекурсия и какова её роль в программировании?
2. Какие существуют типы рекурсий (прямая, обратная, хвостовая)?
3. В чём преимущества и недостатки рекурсивных решений?
4. Что такое шаблон функции и зачем он нужен?
5. Как реализовать обобщённую функцию с помощью шаблонов?

Цель работы:

Изучить понятие рекурсии и научиться строить рекурсивные алгоритмы, а также освоить шаблоны функций для создания обобщённых решений задач.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может

исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 22. Методы сортировки.

Понятие рекурсии и её применение в программировании.

Основные свойства рекурсивных функций: основание и шаг рекурсии.

Примеры рекурсивных алгоритмов: факториал, числа Фибоначчи, Ханойская башня.

Определение шаблона функции и его роль в языке программирования.

Применение шаблонов функций для обобщённого программирования.

Вопросы для самоконтроля:

1. Что такое рекурсия и какова её роль в программировании?
2. Какие существуют типы рекурсий (прямая, непрямая, хвостовая)?
3. В чём преимущества и недостатки рекурсивных решений?
4. Что такое шаблон функции и зачем он нужен?
5. Как реализовать обобщённую функцию с помощью шаблонов?

Цель работы:

Изучить понятие рекурсии и научиться строить рекурсивные алгоритмы, а также освоить шаблоны функций для создания обобщённых решений задач.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью.

Тема 23. Указатели. Динамические переменные.

Понятие сортировки и её назначение в программировании.

Основные методы сортировки: пузырьковая, быстрая, сортировка вставками, выбором, слиянием.

Асимптотическая сложность алгоритмов сортировки: $O(n^2)$, $O(n \log n)$.

Практическое применение сортировок в реальных задачах.

Сравнение различных методов сортировки по критериям эффективности и удобства реализации.

Вопросы для самоконтроля:

1. Что такое сортировка и для чего она применяется?
2. Какие существуют основные методы сортировки?
3. Как оценить эффективность алгоритма сортировки?
4. В чём разница между сортировкой вставками и сортировкой выбором?
5. В каких случаях лучше использовать быструю сортировку, а в каких — сортировку слиянием?

Цель работы:

Изучить и сравнить основные методы сортировки, научиться оценивать их эффективность и выбирать подходящую стратегию сортировки в зависимости от поставленной задачи.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью..

Тема 24. Динамические массивы.

Понятие динамического массива и его отличие от обычного массива.

Выделение и освобождение памяти под динамический массив.

Увеличение и уменьшение размера динамического массива.

Реализация динамических массивов вручную и с использованием STL-контейнеров (std::vector).

Примеры задач, решаемых с помощью динамических массивов.

Вопросы для самоконтроля:

1. Что такое динамический массив и чем он отличается от обычного?
2. Как выделить память под динамический массив?

3. Как изменить размер динамического массива?
4. В каких случаях выгодно использовать динамические массивы?
5. Какие преимущества даёт использование STL-контейнера `vector` по сравнению с ручным созданием динамического массива?

Цель работы:

Изучить работу с динамическими массивами, освоить их реализацию и научиться эффективно изменять их размер в зависимости от потребностей программы.

В работе используется презентация слайдов выполненная в MS Powerpoint.

Критерии оценивания выполнения практического задания:

Оценка «отлично» выставляется, если: работа выполнена в полном объеме с соблюдением необходимой последовательности действий; работа проведена в условиях, обеспечивающих получение правильных результатов и выводов; соблюдены правила техники безопасности; в ответе правильно и аккуратно выполняет все записи, таблицы, рисунки, чертежи, графики, вычисления; правильно выполняет анализ ошибок.

Оценка «хорошо» выставляется, если: работа выполнена правильно с учетом 1-2 мелких погрешностей или 2-3 недочетов, исправленных самостоятельно по требованию преподавателя.

Оценка «удовлетворительно» выставляется, если: работа выполнена правильно не менее чем наполовину, допущены 1-2 погрешности или одна грубая ошибка.

Оценка «неудовлетворительно» выставляется, если: допущены две (и более) грубые ошибки в ходе работы, которые обучающийся не может исправить даже по требованию преподавателя или работа не выполнена полностью..

Типовые тесты

Тест №1

Тема 1. Алгоритм и его свойства.

1. Алгоритм — это...

- А) последовательность действий для решения определённой задачи +
- Б) любая инструкция, написанная человеком
- В) порядок расположения элементов в массиве
- Г) компьютерная программа

2. Свойство алгоритма, означающее, что каждая команда понятна исполнителю и выполнима на практике, называется...

- А) конечность
- Б) массовость
- В) детерминированность
- Г) понятность +

3. Один из важнейших принципов построения алгоритмов заключается в...

- А) бесконечности процесса вычислений
- Б) возможности многократного повторения одних и тех же шагов (цикличности)
- В) случайном порядке выполнения команд
- Г) отсутствию чётко заданных условий завершения

4. Назначение какого свойства алгоритма заключается в однозначности результата при одинаковых исходных данных?

- А) Детерминированность +
- Б) Понятность
- В) Результативность
- Г) Универсальность

5. Процесс разработки алгоритма начинается с этапа...

- А) выбора исполнителя алгоритма
- Б) анализа поставленной задачи и формализации условия +
- В) тестирования готового алгоритма
- Г) написания программного кода

6. Что характеризует свойство конечности алгоритма?

- А) Возможность использования алгоритма различными пользователями
- Б) Наличие четких инструкций для каждого шага
- В) Завершение исполнения алгоритма за конечное число шагов +
- Г) Необходимость использования сложных вычислительных ресурсов

7. Укажите верное утверждение относительно универсальности алгоритма:

- А) Каждый алгоритм применим ко всем возможным данным
- Б) Некоторые алгоритмы могут применяться лишь к ограниченному классу данных +
- В) Любой алгоритм легко адаптируется под любые изменения требований
- Г) Все существующие алгоритмы взаимозаменяемы друг другом

8. Метод пошагового составления плана действий в процессе программирования называется...

- А) циклическим методом
- Б) эвристическим методом
- В) последовательностью операций
- Г) методологией проектирования алгоритмов +

9. О каком свойстве идёт речь, если исполнитель способен выполнить алгоритм независимо от типа устройства или среды выполнения?

- А) Корректность
- Б) Эффективность
- В) Независимость от среды исполнения +
- Г) Минимальная ресурсоемкость

10. Итоговая проверка правильности работы алгоритма осуществляется путем...

- А) теоретического доказательства корректности
- Б) тестирования на реальных примерах +
- В) анализа статистической вероятности ошибок
- Г) консультации эксперта-разработчика

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №2

Тема 2. Понятие сегмента и регистра. Структура ЦП

1. Что такое сегмент в контексте архитектуры компьютера?

- А) Область памяти фиксированного размера
- Б) Область памяти переменного размера +
- В) Устройство ввода-вывода
- Г) Процессорный регистр

2. Какой регистр используется для хранения адреса следующей команды для выполнения?

- А) Регистр данных
- Б) Регистр адреса +
- В) Регистр состояния
- Г) Регистр управления

3. Какой тип сегмента отвечает за хранение кода программы?

- А) Сегмент данных
- Б) Сегмент стека
- В) Сегмент кода +

Г) Сегмент кучи

4. Какой регистр хранит флаги состояния процессора?

- А) Регистр команд
- Б) Регистр флагов +
- В) Регистр указателя стека
- Г) Регистр счётчика команд

5. Какой из перечисленных регистров используется для работы с данными?

- А) Регистр адреса
- Б) Регистр данных +
- В) Регистр состояния
- Г) Регистр кода

6. Какой сегмент используется для хранения временных данных и переменных?

- А) Сегмент кода
- Б) Сегмент стека
- В) Сегмент данных +
- Г) Сегмент адреса

7. Какой из следующих регистров используется для управления потоком выполнения программы?

- А) Регистр данных
- Б) Регистр счётчика команд +
- В) Регистр адреса
- Г) Регистр флагов

8. Как называется структура, которая позволяет процессору работать с сегментами памяти?

- А) Стековая структура
- Б) Сегментация +
- В) Структура управления
- Г) Модуль доступа

9. Какой тип сегмента отвечает за выполнение временных операций и хранения вызовов функций?

- А) Сегмент данных
- Б) Сегмент кода
- В) Сегмент стека +
- Г) Сегмент кучи

10. Какой из регистров отвечает за указание на текущую позицию в стеке?

- А) Регистр данных
- Б) Регистр указателя стека +
- В) Регистр адреса
- Г) Регистр флагов

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №3

Тема 3. Функции языков программирования

1. Что такое функция в языках программирования?

- А) Набор инструкций, выполняющих определённую задачу +
- Б) Переменная, хранящая данные
- В) Команда для выполнения операций
- Г) Структура данных

2. Какой из перечисленных параметров может быть передан функции?

- А) Локальная переменная
- Б) Аргумент +
- В) Глобальная переменная
- Г) Константа

3. Какой из следующих видов функций возвращает значение?

- А) Процедура
- Б) Функция +
- В) Метод
- Г) Подпрограмма

4. Какой термин описывает функцию, которая вызывает саму себя?

- А) Инкапсуляция
- Б) Рекурсия +
- В) Полиморфизм
- Г) Наследование

5. Какой из перечисленных языков программирования не поддерживает

функции как отдельные конструкции?

- А) Python
- Б) Java
- В) HTML +
- Г) C++

6. Какой из следующих параметров функции не является обязательным?

- А) Имя функции
- Б) Возвращаемый тип
- В) Аргументы +
- Г) Тело функции

7. Что такое область видимости функции?

- А) Время выполнения функции
- Б) Место, где функция может быть вызвана +
- В) Размер функции
- Г) Тип данных, возвращаемый функцией

8. Как называется функция, которая не принимает аргументов?

- А) Безвозвратная функция
- Б) Параметрическая функция
- В) Функция без параметров +
- Г) Динамическая функция

9. Какой из следующих аспектов относится к функциональному программированию?

- А) Использование классов
- Б) Отсутствие состояния +
- В) Применение циклов
- Г) Наследование объектов

10. Что такое анонимная функция?

- А) Функция, которая не имеет имени +
- Б) Функция без параметров
- В) Функция, вызывающая другую функцию
- Г) Функция, не возвращающая значение

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии

правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии
правильного ответа студента менее чем на 50% контрольных заданий

Тест №4

Тема 4 Метаязыки описания языков программирования

1. Что такое метаязык?
 - А) Язык, используемый для описания других языков +
 - Б) Язык программирования
 - В) Язык разметки
 - Г) Язык для работы с базами данных
2. Какой из следующих языков является примером метаязыка?
 - А) Python
 - Б) XML +
 - В) Java
 - Г) C++
3. Какова основная цель метаязыка?
 - А) Упрощение написания программ
 - Б) Описание синтаксиса и семантики языков программирования +
 - В) Оптимизация скорости выполнения
 - Г) Создание графических интерфейсов
4. Какой из следующих метаязыков используется для определения синтаксиса языков программирования?
 - А) JSON
 - Б) BNF (Backus-Naur Form) +
 - В) HTML
 - Г) YAML
5. В чем преимущество использования метаязыков?
 - А) Увеличение сложности кода
 - Б) Улучшение взаимодействия между языками +
 - В) Снижение производительности
 - Г) Упрощение работы с аппаратным обеспечением
6. Какой из следующих метаязыков используется для описания структур данных?
 - А) XML +
 - Б) Java
 - В) SQL
 - Г) CSS
7. Что такое грамматика в контексте метаязыков?

- А) Правила, определяющие синтаксис языка +
- Б) Группа функций
- В) Набор данных
- Г) Структура хранения информации

8. Как называется метаязык, используемый для описания форматов данных?

- А) SQL
- Б) JSON +
- В) C#
- Г) PHP

9. Какой из следующих факторов не относится к метаязыкам?

- А) Описание структуры языка
- Б) Определение семантики +
- В) Оптимизация алгоритмов
- Г) Упрощение синтаксиса

10. Как метаязыки могут помочь в разработке компиляторов?

- А) Ускоряют выполнение программ
- Б) Позволяют формализовать правила обработки исходного кода +
- В) Упрощают графический интерфейс
- Г) Увеличивают размер программ

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №5

Тема 5. Виртуальная машина

1. Что такое виртуальная машина (ВМ)?

- А) Физическое устройство для выполнения программ
- Б) Программная эмуляция аппаратного обеспечения +
- В) Программный интерфейс для взаимодействия с пользователем
- Г) Операционная система

2. Какова основная цель виртуальных машин?

- А) Увеличение производительности
- Б) Эмуляция различных архитектур и платформ +
- В) Упрощение программирования
- Г) Улучшение графики

3. Какой из следующих компонентов является частью архитектуры виртуальной машины?

- А) Компилятор
- Б) Интерпретатор +
- В) Система управления базами данных
- Г) Операционная система

4. Какую задачу решает гипервизор в контексте виртуальных машин?

- А) Управление файловой системой
- Б) Управление несколькими виртуальными машинами на одном физическом хосте +
- В) Оптимизация кода
- Г) Обработка пользовательских запросов

5. Какой из следующих типов виртуальных машин работает на уровне операционной системы?

- А) Аппаратная виртуальная машина
- Б) Паравиртуальная машина
- В) Программная виртуальная машина +
- Г) Эмулятор

6. Как называется виртуальная машина, которая используется для выполнения байт-кода?

- А) Java Virtual Machine (JVM) +
- Б) Windows Virtual PC
- В) Hyper-V
- Г) QEMU

7. Какой из следующих аспектов не является преимуществом использования виртуальных машин?

- А) Изоляция приложений
- Б) Меньшие требования к оборудованию +
- В) Легкость в управлении ресурсами
- Г) Возможность тестирования различных сред

8. Какой тип виртуальной машины используется для эмуляции аппаратных компонентов?

- А) Программная виртуальная машина
- Б) Аппаратная виртуальная машина +
- В) Паравиртуальная машина

Г) Гибридная виртуальная машина

9. Какой из следующих сценариев является типичным для применения виртуальных машин?

- А) Разработка мобильных приложений
- Б) Тестирование программного обеспечения в различных средах +
- В) Создание веб-сайтов
- Г) Обработка больших данных

10. Как виртуальные машины помогают в облачных вычислениях?

- А) Увеличивают стоимость услуг
- Б) Позволяют динамически распределять ресурсы +
- В) Усложняют архитектуру приложений
- Г) Ограничивают доступ к данным

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №6

Тема 6 Структура типов языка C++

1. Что такое тип данных в языке C++?

- А) Формат хранения информации +
- Б) Уникальный идентификатор переменной
- В) Способ выполнения операций
- Г) Структура данных

2. Какой из следующих типов данных является встроенным в C++?

- А) `std::vector`
- Б) `int` +
- В) `std::string`
- Г) `std::list`

3. Какой из перечисленных типов данных используется для представления символов?

- А) `int`
- Б) `char` +

- B) float
- Г) bool

4. Какой из следующих типов данных является составным?

- A) int
- Б) bool
- В) struct +
- Г) char

5. Какой из следующих вариантов объявляет массив целых чисел в C++?

- A) int array[10]; +
- Б) array int[10];
- В) int[10] array;
- Г) array<int> myArray;

6. Какой из следующих типов данных используется для представления логических значений?

- A) int
- Б) char
- В) bool +
- Г) float

7. Что такое указатель в языке C++?

- A) Переменная, хранящая адрес другой переменной +
- Б) Переменная, содержащая массив
- В) Функция, которая возвращает значение
- Г) Константа, значение которой не может изменяться

8. Какой из следующих типов данных используется для хранения дробных чисел с одинарной точностью?

- A) int
- Б) double
- В) float +
- Г) long

9. Что такое пользовательский тип данных в C++?

- A) Тип, созданный с помощью стандартных библиотек
- Б) Тип, определенный пользователем с использованием struct или class
- + В) Тип, встроенный в язык
- Г) Тип, который не поддерживается компилятором

10. Какой из следующих типов является ссылкой на другой тип данных?

- A) struct
- Б) reference +

- В) pointer
- Г) typedef

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №7

Тема 7 Целочисленные типы. Символьный тип

1. Какой из следующих типов данных используется для хранения целых чисел в C++?

- A) float
- Б) int +
- В) double
- Г) char

2. Какой из целочисленных типов имеет наибольший диапазон значений?

- A) short
- Б) int
- В) long +
- Г) char

3. Какой из следующих типов представляет отрицательные и положительные числа?

- A) unsigned int
- Б) int +
- В) short
- Г) char

4. Какой из типов данных используется для хранения целых чисел без знака?

- A) signed int
- Б) unsigned int +
- В) long
- Г) float

5. Какой из следующих типов данных имеет размер в 2 байта на большинстве платформ?

- A) int
- Б) short +
- В) long
- Г) char

6. Какой тип данных в C++ используется для представления одиночного символа?

- A) string
- Б) char +
- В) wchar
- Г) int

7. Какой из следующих типов используется для хранения широкой символьной информации?

- A) char
- Б) unsigned char
- В) wchar +
- Г) int

8. Как объявить переменную символьного типа в C++?

- A) char myChar; +
- Б) myChar char;
- В) char = myChar;
- Г) myChar: char;

9. Какой из следующих символов используется для обозначения строки в C++?

- A) ''
- Б) " " +
- В) ``
- Г) []

10. Какой метод используется для получения длины строки в C++?

- A) strlen() +
- Б) length()
- В) size()
- Г) count()

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий

- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №8

Тема 8 Булевский (логический) тип. Вещественные типы.

1. Какой тип данных в C++ используется для представления логических значений?
 - А) int
 - Б) float
 - В) bool +
 - Г) char
2. Какие два значения может принимать булевский тип?
 - А) 0 и 1
 - Б) true и false +
 - В) yes и no
 - Г) on и off
3. Как объявить переменную булевского типа в C++?
 - А) bool myBool; +
 - Б) myBool bool;
 - В) bool = myBool;
 - Г) myBool: bool;
4. Какое значение будет присвоено переменной типа bool при инициализации с 1?
 - А) false
 - Б) true +
 - В) 0
 - Г) undefined
5. Какой из следующих операторов используется для логического И в C++?
 - А) ||
 - Б) && +
 - В) !
 - Г) ==
6. Какой тип данных в C++ используется для хранения дробных чисел с одинарной точностью?

- A) double
- Б) float +
- В) long
- Г) int

7. Какой из следующих типов данных предоставляет двойную точность для вещественных чисел?

- A) float
- Б) double +
- В) decimal
- Г) short

8. Какое значение будет присвоено переменной типа float, если она инициализируется как float myFloat = 5.5?

- A) 5
- Б) 5.5 +
- В) 6
- Г) 5.0

9. Как объявить переменную вещественного типа в C++?

- A) float myNumber; +
- Б) myNumber float;
- В) float = myNumber;
- Г) myNumber: float;

10. Какой из следующих методов используется для округления вещественного числа в C++?

- A) round() +
- Б) ceil()
- В) floor()
- Г) abs()

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №9

Тема 9 Описание переменных и констант. Арифметические операции.

1. Что такое переменная в языке C++?
 - А) Константа, значение которой не может изменяться
 - Б) Именованная область памяти для хранения данных +
 - В) Функция, которая возвращает значение
 - Г) Структура, которая содержит информацию
2. Как объявить переменную целочисленного типа в C++?
 - А) `int myVar; +`
 - Б) `myVar int;`
 - В) `int = myVar;`
 - Г) `myVar: int;`
3. Какой из следующих вариантов правильно объявляет константу в C++?
 - А) `const int MYCONST = 10; +`
 - Б) `int const MYCONST = 10;`
 - В) `#define MYCONST 10;`
 - Г) `const MYCONST = 10;`
4. Какой оператор используется для присвоения значения переменной?
 - А) `==`
 - Б) `:=`
 - В) `= +`
 - Г) `=>`
5. Какой из следующих способов инициализирует переменную сразу при объявлении?
 - А) `int myVar; myVar = 5;`
 - Б) `int myVar = 5; +`
 - В) `myVar: int = 5;`
 - Г) `int myVar: 5;`
6. Какой из следующих операторов используется для сложения двух чисел в C++?
 - А) `-`
 - Б) `++`
 - В) `*`
 - Г) `/`
7. Какой результат будет у выражения `'5 / 2'` при использовании целочисленных типов?
 - А) 2.5
 - Б) 2 +

- В) 3
- Г) 2.0

8. Какой оператор используется для получения остатка от деления в C++?

- А) /
- Б) % +
- В) **
- Г) //

9. Какой из следующих операторов используется для вычитания?

- А) +
- Б) - +
- В) *
- Г) /

10. Какой результат будет у выражения `++x`, если `x` равно 5?

- А) 4
- Б) 5
- В) 6 +
- Г) 7

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №10

Тема 10 Поразрядные операции языка C++. Операторы языка.

1. Какой оператор используется для поразрядного И в C++?

- А) & +
- Б) |
- В) ^
- Г) ~

2. Какой результат будет у выражения `5 & 3`? (в двоичном представлении 5 = 101, 3 = 011)

- A) 0
- Б) 1
- В) 3 +
- Г) 5

3. Какой оператор используется для поразрядного ИЛИ в C++?

- A) &
- Б) | +
- В) ^
- Г) ~

4. Какой результат будет у выражения `5 | 3`? (в двоичном представлении $5 = 101$, $3 = 011$)

- A) 0
- Б) 5
- В) 7 +
- Г) 3

5. Какой оператор используется для поразрядного исключающего ИЛИ (XOR)?

- A) &
- Б) |
- В) ^ +
- Г) ~

6. Какой из следующих операторов используется для равенства в C++?

- A) =
- Б) == +
- В) ===
- Г) :=

7. Какой оператор используется для инкремента значения переменной на 1?

- A) ++ +
- Б) +=
- В) --
- Г) =+

8. Какой оператор используется для декремента значения переменной на 1?

- A) -- +
- Б) -=
- В) ++
- Г) -=

9. Какой из следующих операторов используется для сложения и присваивания?

- А) +
- Б) += +
- В) =+
- Г) +:

10. Какой оператор используется для логического отрицания?

- А) ! +
- Б) ~
- В) not
- Г) ^

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №11

Тема 11 Средства ввода-вывода. Условный оператор

1. Какой заголовочный файл необходимо подключить для использования стандартных средств ввода-вывода в C++?

- А) <iostream> +
- Б) <stdio.h>
- В) <inputoutput>
- Г) <cstdio>

2. Какой объект используется для вывода данных на экран в C++?

- А) cin
- Б) cout +
- В) printf
- Г) output

3. Какой оператор используется для вывода данных с помощью cout?

- А) << +
- Б) >>
- В) +

Г) =

4. Какой объект используется для ввода данных с клавиатуры в C++?

- А) cin +
- Б) input
- В) scanf
- Г) read

5. Какой из следующих методов считывает строку из ввода в C++?

- А) cin >> myString; +
- Б) cin << myString;
- В) read(myString);
- Г) scanf(myString);

Условный оператор

6. Какой оператор в C++ используется для условных выражений?

- А) if +
- Б) switch
- В) case
- Г) while

7. Какой из следующих синтаксисов правильный для оператора if в C++?

- А) if (x > 0) { / действия / } +
- Б) if x > 0 { / действия / }
- В) if { x > 0 } { / действия / }
- Г) if (x > 0) / действия /

8. Какой оператор используется для проверки нескольких условий в одном выражении?

- А) || +
- Б) &&
- В) !
- Г) ==

9. Какой из следующих вариантов является правильным использованием оператора switch?

- А) switch (x) { case 1: / действия / break; } +
- Б) switch x { case 1: / действия / }
- В) switch (x) { case 1: / действия / }
- Г) switch { x }

10. Какой результат будет у следующего кода, если x = 10:

```
if (x > 5)
    cout << "More than 5";
```

else

cout << "5 or less";

A) More than 5 +

Б) 5 or less

В) 10

Г) Ошибка компиляции

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №12

Тема 12 Оператор выбора. Операторы цикла.

1. Какой оператор в C++ используется для выбора между несколькими вариантами?
A) if
Б) switch +
В) for
Г) while
2. Какой из следующих синтаксисов является правильным для оператора switch?
A) switch (x) { case 1: / действия / break; } +
Б) switch x { case 1: / действия / }
В) switch (x) { case 1: / действия / }
Г) switch { x }
3. Какой оператор используется для выхода из блока switch?
A) exit
Б) return
В) break +
Г) stop
4. Как правильно использовать оператор case в switch?
A) case 1: / действия / +
Б) case (1) { / действия / }

- В) case 1; / действия /
- Г) case 1: -> / действия /

5. Что произойдет, если не использовать оператор break в switch?

- А) Программа завершится с ошибкой
- Б) Выполнение перейдет к следующему case +
- В) Программа будет работать медленнее
- Г) Ничего не произойдет

6. Какой оператор используется для создания цикла с известным количеством итераций?

- А) while
- Б) do-while
- В) for +
- Г) repeat

7. Какой из следующих синтаксисов правильный для цикла for?

- А) for (int i = 0; i < 10; i++) { / действия / } +
- Б) for int i = 0; i < 10; i++ { / действия / }
- В) for (i = 0; i < 10; i++) { / действия / }
- Г) for (int i; i < 10; i++) { / действия / }

8. Какой оператор используется в цикле do-while для проверки условия?

- А) if
- Б) while +
- В) for
- Г) switch

9. Какой результат будет у следующего кода, если `i` инициализировано значением 0:

```
while (i < 3) {  
    cout << i;  
    i++;  
}
```

- А) 0123
- Б) 012 +
- В) 123
- Г) 0

10. Какой из следующих циклов выполнится хотя бы один раз, даже если условие изначально ложно?

- А) while
- Б) for

- В) do-while +
- Г) foreach

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №13

Тема 13 Операторы перехода. Обработка последовательностей.

1. Какой оператор используется для немедленного выхода из цикла или блока кода?

- А) continue
- Б) break +
- В) return
- Г) exit

2. Какой оператор позволяет пропустить текущую итерацию цикла и перейти к следующей?

- А) break
- Б) continue +
- В) exit
- Г) skip

3. Какой оператор используется для завершения выполнения функции и возврата значения?

- А) exit
- Б) return +
- В) break
- Г) end

4. Какой из следующих операторов можно использовать для перехода к определенной метке в коде?

- А) goto +
- Б) label
- В) break
- Г) continue

5. Какой из следующих случаев может привести к неуправляемому поведению программы при использовании оператора goto?

- А) Переход к метке, находящейся в другом блоке
- Б) Переход к метке, которая не существует
- В) Бесконечный цикл из-за неправильного использования goto +
- Г) Все перечисленные

6. Какой стандартный контейнер C++ используется для хранения последовательности элементов, доступ к которым осуществляется по индексу?

- А) list
- Б) queue
- В) vector +
- Г) set

7. Какой метод позволяет добавить элемент в конец вектора?

- А) add
- Б) pushback +
- В) insert
- Г) append

8. Какой метод используется для удаления последнего элемента вектора?

- А) remove
- Б) erase
- В) popback +
- Г) delete

9. Какой из следующих контейнеров позволяет хранить элементы в упорядоченном виде и не допускает дубликатов?

- А) vector
- Б) list
- В) set +
- Г) deque

10. Какой метод используется для сортировки элементов вектора?

- А) sort +
- Б) order
- В) arrange
- Г) organize

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий

- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №14

Тема 14 Итерационные циклы. Программирование вложенных. Циклов.

1. Какой из следующих операторов используется для создания бесконечного цикла?
 - А) while (true) +
 - Б) for (1; 1; 1)
 - В) do { } while (1);
 - Г) all of the above
2. Какой из следующих циклов будет выполняться как минимум один раз?
 - А) while
 - Б) for
 - В) do-while +
 - Г) foreach
3. Как правильно записать цикл for, который проходит от 0 до 4?
 - А) for (int i = 0; i < 5; i++) { / действия / } +
 - Б) for (int i = 0; i <= 4; i++) { / действия / }
 - В) for (int i = 0; i < 4; i++) { / действия / }
 - Г) for (int i = 1; i < 5; i++) { / действия / }
4. Что произойдет, если в условии цикла while использовать всегда истинное выражение?
 - А) Цикл выполнится один раз
 - Б) Цикл выполнится дважды
 - В) Цикл выполнится бесконечно +
 - Г) Цикл не выполнится
5. Какой из следующих операторов может быть использован для прерывания выполнения цикла?
 - А) continue
 - Б) break +
 - В) exit
 - Г) stop
6. Какой из следующих примеров представляет собой вложенный цикл?
 - А) for (int i = 0; i < 5; i++) {

```

    for (int j = 0; j < 5; j++) {
        // действия
    }
} +
Б) while (i < 5) {
    // действия
    i++;
}
В) do {
    // действия
} while (i < 5);
Г) for (int i = 0; i < 5; i++) {
    // действия
}

```

7. Какое количество итераций выполнится в следующем коде, если `n = 3`?

```

for (int i = 0; i < n; i++) {
    for (int j = 0; j < n; j++) {
        // действия
    }
}

```

- А) 3
- Б) 6 +
- В) 9
- Г) 12

8. Какой из следующих методов можно использовать для выхода из вложенного цикла?

- А) break +
- Б) continue
- В) exit
- Г) return

9. Что произойдет, если не использовать оператор break в вложенном цикле, и условие останется истинным?

- А) Внешний цикл завершится
- Б) Внутренний цикл завершится
- В) Программа будет выполнена без ошибок +
- Г) Внешний цикл будет продолжаться до тех пор, пока условие внутреннего цикла истинно

10. Какой из следующих примеров кода демонстрирует использование вложенного цикла для создания таблицы умножения?

```

+A) for (int i = 1; i <= 10; i++) {
    for (int j = 1; j <= 10; j++) {
        cout << i <i> j << " ";
    }
}

```

```

    }
    cout << endl;
}
Б) while (i <= 10) {
    while (j <= 10) {
        cout << i << " ";
        j++;
    }
    i++;
}
В) do {
    cout << i * j << " ";
} while (j <= 10);
Г) for (int i = 1; i <= 10; i++) {
    cout << i << endl;
}

```

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №15

Тема 15 Регулярные типы. Обработка одномерных массивов.

1. Какой из следующих типов данных является целым числом в C++?
 - А) float
 - Б) double
 - В) int +
 - Г) char
2. Какой из следующих типов данных используется для хранения символов?
 - А) string
 - Б) char +
 - В) int
 - Г) bool

3. Какой из следующих типов данных может хранить значения с плавающей точкой?

- A) int
- Б) float +
- В) char
- Г) bool

4. Какой из следующих типов данных используется для хранения логических значений (истина или ложь)?

- A) int
- Б) bool +
- В) char
- Г) float

5. Какой из приведенных типов данных имеет наибольший диапазон значений?

- A) short
- Б) int
- В) long +
- Г) char

6. Как правильно объявить одномерный массив целых чисел размером 10?

- A) int array[10]; +
- Б) int array(10);
- В) array[10] int;
- Г) int[10] array;

7. Какой оператор используется для доступа к элементу массива по индексу?

- A) .
- Б) [] +
- В) ()
- Г) {}

8. Какой из следующих способов позволяет инициализировать массив целых чисел значениями от 1 до 5?

- A) int array[] = {1, 2, 3, 4, 5}; +
- Б) int array(5) = {1, 2, 3, 4, 5};
- В) int array[5] = 1, 2, 3, 4, 5;
- Г) int array = {1, 2, 3, 4, 5};

9. Какой метод используется для получения размера одномерного массива в C++?

- A) sizeof(array) +

- Б) length(array)
- В) array.size()
- Г) count(array)

10. Какой из следующих примеров кода правильно выводит все элементы массива?

- + А)

```
for (int i = 0; i < 10; i++) {  
    cout << array[i] << " ";  
}
```
- Б)

```
for (int i = 1; i <= 10; i++) {  
    cout << array[i] << " ";  
}
```
- В)

```
for (int i = 0; i <= 10; i++) {  
    cout << array[i] << " ";  
}
```
- Г)

```
for (int i = 0; i < 10; i++) {  
    cout << array(i) << " ";  
}
```

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №16

Тема 16 Классы задач по обработке массивов. Задачи 1-ого и 2-го класса.

1. Какой из следующих примеров относится к задаче 1-го класса по обработке массивов?

- А) Найти максимальный элемент в массиве +
- Б) Отсортировать массив по возрастанию
- В) Сложить все элементы массива и вывести сумму
- Г) Найти количество уникальных элементов в массиве

2. Какой из следующих примеров относится к задаче 2-го класса по обработке массивов?

- А) Найти среднее значение элементов массива

- Б) Найти индекс первого вхождения элемента в массиве +
- В) Объединить два отсортированных массива в один
- Г) Вывести элементы массива в обратном порядке

3. Какой из следующих алгоритмов можно использовать для поиска максимального элемента в массиве?

- А) Сортировка выбором
- Б) Линейный поиск +
- В) Бинарный поиск
- Г) Сортировка вставками

4. Какой из следующих способов можно использовать для нахождения суммы элементов массива?

- А) Использовать цикл for +
- Б) Использовать рекурсию
- В) Использовать цикл while
- Г) Все перечисленные варианты

5. Какой из приведенных примеров кода правильно реализует задачу 1-го класса — нахождение максимального элемента в массиве?

+ А)

```
int max = array[0];
for (int i = 1; i < n; i++) {
    if (array[i] > max) {
        max = array[i];
    }
}
```

Б)

```
int max = 0;
for (int i = 0; i < n; i++) {
    if (array[i] < max) {
        max = array[i];
    }
}
```

В)

```
int max = array[0];
for (int i = 0; i < n; i++) {
    max = array[i];
}
```

Г)

```
int max = array[n - 1];
for (int i = n - 2; i >= 0; i--) {
    if (array[i] > max) {
        max = array[i];
    }
}
```

}

6. Какой из следующих методов можно использовать для нахождения среднего значения элементов массива?

- А) Сложить все элементы и разделить на количество элементов +
- Б) Найти максимальный и минимальный элементы и усреднить их
- В) Использовать бинарный поиск
- Г) Все перечисленные варианты

7. Какой из следующих примеров демонстрирует задачу 2-го класса — нахождение индекса первого вхождения элемента в массиве?

+ А)

```
int index = -1;
for (int i = 0; i < n; i++) {
    if (array[i] == value) {
        index = i;
        break;
    }
}
```

Б)

```
int index = 0;
while (index < n) {
    if (array[index] == value) {
        break;
    }
    index++;
}
```

В)

```
int index = -1;
for (int i = 0; i < n; i++) {
    if (array[i] == value) {
        index = i;
    }
}
```

Г)

```
int index = -1;
for (int i = n - 1; i >= 0; i--) {
    if (array[i] == value) {
        index = i;
        break;
    }
}
```

8. Какой из следующих алгоритмов можно использовать для объединения двух отсортированных массивов в один?

- А) Сортировка вставками
- Б) Сортировка выбором
- В) Алгоритм слияния +
- Г) Линеиный поиск

9. Какое из следующих утверждений верно для задач 1-го класса?

- А) Они требуют сложных алгоритмов
- Б) Они требуют простых вычислений и операций с массивами +
- В) Они всегда требуют сортировки
- Г) Они не могут быть решены с помощью циклов

10. Какой из следующих примеров кода правильно реализует задачу 2-го класса — н

ахождение количества уникальных элементов в массиве?

+ А)

```
int count = 0;
for (int i = 0; i < n; i++) {
    bool isUnique = true;
    for (int j = 0; j < i; j++) {
        if (array[i] == array[j]) {
            isUnique = false;
            break;
        }
    }
    if (isUnique) {
        count++;
    }
}
```

Б)

```
int count = 0;
for (int i = 0; i < n; i++) {
    count++;
}
```

В)

```
int count = 0;
for (int i = 0; i < n; i++) {
    if (array[i] != array[i + 1]) {
        count++;
    }
}
```

Г)


```
int count = 0;
for (int i = 0; i < n; i++) {
    count += array[i];
}
```

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №17

Тема 17 Классы задач по обработке массивов. Задачи 3-ого и 4-го класса.

1. Какой из следующих примеров относится к задаче 3-го класса по обработке массивов?
 - А) Найти количество четных элементов в массиве +
 - Б) Найти максимальный элемент в массиве
 - В) Отсортировать массив по убыванию
 - Г) Найти сумму элементов массива
2. Какой из следующих примеров относится к задаче 4-го класса по обработке массивов?
 - А) Определить, является ли массив палиндромом +
 - Б) Найти индекс последнего вхождения элемента в массиве
 - В) Найти среднее значение элементов массива
 - Г) Найти количество уникальных элементов в массиве
3. Какой из следующих алгоритмов можно использовать для проверки, является ли массив палиндромом?
 - А) Сравнить элементы с начала и конца +
 - Б) Отсортировать массив
 - В) Найти максимальный элемент
 - Г) Подсчитать количество элементов
4. Какой из следующих методов можно использовать для нахождения количества четных элементов в массиве?
 - А) Использовать цикл for +

- Б) Использовать рекурсию
- В) Использовать массив для хранения четных элементов
- Г) Все перечисленные варианты

5. Какой из приведенных примеров кода правильно реализует задачу 3-го класса — нахождение количества четных элементов в массиве?

+А)
int count = 0;
for (int i = 0; i < n; i++) {
 if (array[i] % 2 == 0) {
 count++;
 }
}

Б)
int count = 0;
for (int i = 0; i < n; i++) {
 if (array[i] % 2 != 0) {
 count++;
 }
}

В)
int count = 0;
for (int i = 0; i < n; i++) {
 count += array[i];
}

Г)
int count = 0;
for (int i = 0; i < n; i++) {
 if (array[i] < 0) {
 count++;
 }
}

6. Какой из следующих способов можно использовать для определения, является ли массив палиндромом?

- А) Сравнить элементы от начала и конца массива +
- Б) Найти сумму всех элементов
- В) Отсортировать массив и сравнить его с оригиналом
- Г) Подсчитать количество четных и нечетных элементов

7. Какой из следующих примеров демонстрирует задачу 4-го класса — проверку на палиндром?

+ А)
bool isPalindrome = true;
for (int i = 0; i < n / 2; i++) {
 if (array[i] != array[n - 1 - i]) {

```

        isPalindrome = false;
        break;
    }
}
Б)
bool isPalindrome = false;
for (int i = 0; i < n; i++) {
    if (array[i] == array[n - 1]) {
        isPalindrome = true;
        break;
    }
}

```

```

В)
bool isPalindrome = true;
for (int i = 0; i < n; i++) {
    if (array[i] == array[i + 1]) {
        isPalindrome = false;
        break;
    }
}

```

```

Г)
bool isPalindrome = true;
for (int i = 0; i < n; i++) {
    if (array[i] == 0) {
        isPalindrome = false;
        break;
    }
}

```

8. Какой из следующих методов можно использовать для нахождения суммы всех элементов массива?

- А) Использовать цикл for +
- Б) Использовать только рекурсию
- В) Использовать встроенные функции
- Г) Все перечисленные варианты

9. Какое из следующих утверждений верно для задач 3-го класса?

- А) Они требуют сложных алгоритмов
- Б) Они требуют сложных вычислений и манипуляций с массивами
- В) Они могут включать условия и циклы для обработки данных +
- Г) Они всегда требуют сортировки

10. Какой из следующих примеров кода правильно реализует задачу 4-го класса — проверку, является ли массив палиндромом?

```

+A)
bool isPalindrome = true;
for (int i = 0; i < n / 2; i++) {
    if (array[i] != array[n - 1 - i]) {
        isPalindrome = false;
        break;
    }
}
}

Б)
bool isPalindrome = true;
if (n % 2 == 0) {
    for (int i = 0; i < n / 2; i++) {
        if (array[i] != array[n - 1 - i]) {
            isPalindrome = false;
            break;
        }
    }
}
}

В)
bool isPalindrome = false;
for (int i = 0; i < n; i++) {
    if (array[i] == array[n - 1]) {
        isPalindrome = true;
        break;
    }
}
}

Г)
bool isPalindrome = true;
for (int i = 0; i < n; i++) {
    if (array[i] == 0) {
        isPalindrome = false;
        break;
    }
}
}

```

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии

правильного ответа студента менее чем на 50% контрольных заданий

Тест №18

Тема 18 Двумерные массивы

1. Как вы можете представить двумерный массив?
 - А) Как таблицу с строками и столбцами +
 - Б) Как одномерный список
 - В) Как набор отдельных значений
 - Г) Как график

2. Какой из следующих примеров показывает, как можно объявить двумерный массив, который хранит оценки студентов по предметам?
 - А) `int grades[5][3];`` +
 - Б) `int grades(5, 3);``
 - В) `int grades[3][5];``
 - Г) `int grades[5, 3];``

3. Что происходит, если вы пытаетесь получить доступ к элементу массива, используя индекс, который выходит за пределы его размера?
 - А) Программа завершится успешно
 - Б) Вы получите случайное значение
 - В) Программа может аварийно завершиться или выдать ошибку +
 - Г) Элемент автоматически создастся

4. Какой способ подходит для инициализации двумерного массива с начальными значениями?
 - А) Можно указать значения в фигурных скобках в виде таблицы +
 - Б) Можно инициализировать только нулями
 - В) Можно использовать только один массив
 - Г) Нельзя инициализировать двумерный массив

5. Как вы можете найти сумму всех элементов в двумерном массиве?
 - А) Использовать один цикл
 - Б) Использовать вложенные циклы +
 - В) Только с помощью рекурсии
 - Г) Сложить только элементы по диагонали

6. Как вы можете вывести все элементы двумерного массива на экран?
 - А) Использовать один цикл `for` +
 - Б) Использовать два вложенных цикла `for`
 - В) Записать все элементы в строку
 - Г) Нельзя выводить элементы двумерного массива

7. Какой метод может быть использован для транспонирования двумерного массива?

- А) Переставить строки и столбцы местами +
- Б) Упорядочить элементы по возрастанию
- В) Найти сумму всех элементов
- Г) Сложить соответствующие элементы двух массивов

8. Что происходит, когда вы пытаетесь изменить размер двумерного массива после его создания?

- А) Размер автоматически изменится
- Б) Это невозможно, нужно создавать новый массив +
- В) Программа выдаст ошибку
- Г) Изменится только первый элемент

9. Как можно проверить, является ли двумерный массив квадратным (одинаковое количество строк и столбцов)?

- А) Сравнить количество строк и столбцов +
- Б) Проверить, что все элементы равны
- В) Посчитать сумму всех элементов
- Г) Это невозможно проверить

10. Как вы можете найти максимальный элемент в двумерном массиве?

- А) Сравнить элементы в каждой строке +
- Б) Отсортировать массив
- В) Найти только максимальный элемент в первой строке
- Г) Сложить все элементы и выбрать наибольший

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №19

Тема 19 Функции. Прототип функции. Параметры функции.

1. Что такое функция в языке C++?

- А) Это блок кода, который выполняет определенную задачу +
- Б) Это просто переменная
- В) Это способ организации данных

Г) Это тип данных

2. Каковы основные преимущества использования функций?

- А) Упрощение кода и повторное использование +
- Б) Увеличение размера программы
- В) Упрощение компиляции
- Г) Сложность в понимании кода

3. Что такое прототип функции?

- А) Это полное определение функции с кодом
- Б) Это объявление функции, которое показывает, что она существует +
- В) Это просто имя функции
- Г) Это описание параметров функции без их типов

4. Как правильно записать прототип функции, которая возвращает целое число и принимает два целых числа в качестве параметров?

- А) ``int add(int a, int b);`` +
- Б) ``void add(int, int);``
- В) ``float add(int a, int b);``
- Г) ``int add(a, b);``

5. Как вы вызываете функцию в программе?

- А) Путем указания ее имени и передачи необходимых аргументов +
- Б) Путем указания только имени функции
- В) Путем написания самой функции повторно
- Г) Путем создания нового прототипа

6. Что произойдет, если вы вызовете функцию, не передав ей необходимые параметры?

- А) Программа завершится успешно
- Б) Будет использовано значение по умолчанию
- В) Программа выдаст ошибку при компиляции +
- Г) Функция выполнится с пропущенными значениями

7. Какой из следующих способов позволяет передать аргументы в функцию по ссылке, чтобы изменения в функции отразились на переданных аргументах?

- А) Использовать обычные параметры
- Б) Использовать указатели +
- В) Использовать только константные параметры
- Г) Нельзя передавать аргументы по ссылке

8. Какой тип данных может возвращать функция?

- А) Только целые числа
- Б) Только символы

- В) Любой тип данных, включая пользовательские типы +
- Г) Функции не могут возвращать значения

9. Как вы можете передать массив в функцию?

- А) Передать его как указатель +
- Б) Передать его как значение
- В) Нельзя передать массив в функцию
- Г) Передать его как строку

10. Как правильно вызвать функцию, которая принимает два целых числа?

- А) ``add(5, 10);`` +
- Б) ``add(5);``
- В) ``add(5, 10, 15);``
- Г) ``add(a, b);``

Критерии оценивания тестовых заданий для текущего контроля

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

Тест №20

Тема 20 Локальные и глобальные переменные. Область видимости переменных

1. Что такое глобальная переменная в C++?

- А) Переменная, которая доступна только внутри одной функции
- Б) Переменная, которая доступна во всей программе после ее объявления +
- В) Переменная, которая может использоваться только в одном файле
- Г) Переменная, которая не может быть изменена

2. Какова область видимости локальной переменной?

- А) Вся программа
- Б) Вся функция, в которой она объявлена +
- В) Только в том блоке кода, где она была объявлена
- Г) Только в файле, где она объявлена

3. Как можно объявить глобальную переменную?
- А) Внутри функции
 - Б) Вне всех функций +
 - В) Внутри класса
 - Г) Внутри метода
4. Что произойдет, если локальная переменная имеет то же имя, что и глобальная переменная?
- А) Локальная переменная будет проигнорирована
 - Б) Локальная переменная будет скрывать глобальную переменную +
 - В) Программа завершится с ошибкой
 - Г) Глобальная переменная будет изменена
5. Как вы можете получить доступ к глобальной переменной внутри функции, если в ней объявлена локальная переменная с тем же именем?
- А) Просто использовать имя переменной
 - Б) Использовать ключевое слово ``global``
 - В) Использовать оператор разрешения области видимости ``::`` +
 - Г) Это невозможно
6. Какова основная причина использования локальных переменных?
- А) Упрощение кода и предотвращение конфликтов имен +
 - Б) Увеличение области видимости переменных
 - В) Снижение производительности программы
 - Г) Упрощение работы с глобальными переменными
7. Какой из следующих примеров правильно демонстрирует использование глобальной и локальной переменной?
- ```
int globalVar = 10; // Глобальная переменная

void function() {
 int localVar = 5; // Локальная переменная
 // Код здесь
} +
```
- А) Это неправильный пример
  - Б) Это правильный пример +
  - В) Глобальная переменная недоступна
  - Г) Локальная переменная доступна вне функции
8. Что произойдет, если вы попытаетесь использовать локальную переменную за пределами ее области видимости?
- А) Программа завершится успешно
  - Б) Будет использовано значение по умолчанию
  - В) Программа выдаст ошибку компиляции +

Г) Локальная переменная будет автоматически преобразована в глобальную

9. В каком случае использование глобальных переменных может быть нежелательным?

- А) Когда необходимо использовать их в разных функциях
- Б) Когда нужно избежать конфликтов имен и сделать код более читаемым +
- В) Когда они необходимы для хранения состояния программы
- Г) Когда нужно передавать данные между функциями

10. Как можно улучшить читабельность кода при использовании переменных?

- А) Использовать только глобальные переменные
- Б) Использовать понятные имена переменных и локальные переменные +
- В) Сокращать имена переменных
- Г) Игнорировать область видимости

11. Какой из следующих вариантов является верным утверждением о области видимости глобальных переменных?

- А) Глобальные переменные доступны только в файле, где они объявлены
- Б) Глобальные переменные доступны во всех функциях после их объявления +
- В) Глобальные переменные не могут быть изменены
- Г) Глобальные переменные имеют ту же область видимости, что и локальные

### **Критерии оценивания тестовых заданий для текущего контроля**

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

### **Тест №21**

#### **Тема 21 . Рекурсия. Шаблоны функций**

1. Что такое рекурсия в C++?

- А) Это функция, которая вызывает другую функцию
- Б) Это функция, которая вызывает саму себя +
- В) Это способ организации данных
- Г) Это метод компиляции программы

2. Какой из следующих примеров является правильной рекурсивной функцией для вычисления факториала числа?

```
int factorial(int n) {
 if (n == 0) return 1; // Базовый случай
 return n * factorial(n - 1); // Рекурсивный случай
} +
```

- А) Это неправильный пример
- Б) Это правильный пример +
- В) Базовый случай отсутствует
- Г) Функция не рекурсивная

3. Каковы основные преимущества использования рекурсии?

- А) Упрощение написания кода для некоторых задач +
- Б) Увеличение размера программы
- В) Упрощение компиляции
- Г) Сложность в понимании кода

4. Какой из следующих случаев является примером базового случая в рекурсии?

- А) Условие, при котором рекурсивные вызовы прекращаются +
- Б) Условие, при котором функция вызывает саму себя
- В) Условие, при котором функция возвращает значение
- Г) Условие, при котором функция изменяет глобальные переменные

5. Какой из следующих вариантов может привести к ошибке переполнения стека при использовании рекурсии?

- А) Если базовый случай отсутствует +
- Б) Если функция возвращает значение
- В) Если рекурсивные вызовы имеют разные параметры
- Г) Если функция не вызывает другую функцию

6. Что такое шаблон функции в C++?

- А) Это функция, которая может принимать только один тип данных
- Б) Это функция, которая может работать с разными типами данных +
- В) Это обычная функция без параметров
- Г) Это функция, которая обязательно должна возвращать значение

7. Какой из следующих примеров демонстрирует вызов шаблона функции?

```
int result = add(5, 10); // Вызов с целыми числами +
double sum = add(5.5, 2.3); // Вызов с вещественными числами +
```

- А) Это неправильный пример
- Б) Это правильный пример +
- В) Шаблоны не поддерживают перегрузку
- Г) Вызов функции с неправильными типами данных

8. Какой из следующих случаев является примером рекурсивного подхода?

- А) Поиск максимального элемента в массиве
- Б) Сортировка массива
- В) Вычисление чисел Фибоначчи +
- Г) Сложение элементов массива

9. Что происходит, если шаблон функции не может быть инстанцирован для определенного типа?

- А) Программа завершится успешно
- Б) Программа выдаст ошибку компиляции +
- В) Компилятор автоматически создаст другой шаблон
- Г) Шаблон будет проигнорирован

10. Какой из следующих вариантов является основной причиной использования шаблонов функций?

- А) Они позволяют писать код, который работает только с одним типом данных
- Б) Они упрощают код и повышают его гибкость, позволяя работать с различными типами данных +
- В) Они увеличивают время компиляции
- Г) Они делают код сложнее для понимания

### **Критерии оценивания тестовых заданий для текущего контроля**

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

### **Тест №22**

#### **Тема 22 Методы сортировки**

1. Что такое рекурсия в C++?
  - А) Это функция, которая вызывает другую функцию
  - Б) Это функция, которая вызывает саму себя +
  - В) Это способ организации данных
  - Г) Это метод компиляции программы
  
2. Какой из следующих примеров является правильной рекурсивной функцией для вычисления факториала числа?

```
int factorial(int n) {
 if (n == 0) return 1; // Базовый случай
 return n * factorial(n - 1); // Рекурсивный случай
} +
```

  - А) Это неправильный пример
  - Б) Это правильный пример +
  - В) Базовый случай отсутствует
  - Г) Функция не рекурсивная
  
3. Каковы основные преимущества использования рекурсии?
  - А) Упрощение написания кода для некоторых задач +
  - Б) Увеличение размера программы
  - В) Упрощение компиляции
  - Г) Сложность в понимании кода
  
4. Какой из следующих случаев является примером базового случая в рекурсии?
  - А) Условие, при котором рекурсивные вызовы прекращаются +
  - Б) Условие, при котором функция вызывает саму себя
  - В) Условие, при котором функция возвращает значение
  - Г) Условие, при котором функция изменяет глобальные переменные
  
5. Какой из следующих вариантов может привести к ошибке переполнения стека при использовании рекурсии?
  - А) Если базовый случай отсутствует +
  - Б) Если функция возвращает значение
  - В) Если рекурсивные вызовы имеют разные параметры
  - Г) Если функция не вызывает другую функцию
  
6. Что такое шаблон функции в C++?
  - А) Это функция, которая может принимать только один тип данных
  - Б) Это функция, которая может работать с разными типами данных +
  - В) Это обычная функция без параметров
  - Г) Это функция, которая обязательно должна возвращать значение

7. Какой из следующих примеров демонстрирует вызов шаблона функции?

int result = add(5, 10); // Вызов с целыми числами +

double sum = add(5.5, 2.3); // Вызов с вещественными числами +

- А) Это неправильный пример
- Б) Это правильный пример +
- В) Шаблоны не поддерживают перегрузку
- Г) Вызов функции с неправильными типами данных

8. Какой из следующих случаев является примером рекурсивного подхода?

- А) Поиск максимального элемента в массиве
- Б) Сортировка массива
- В) Вычисление чисел Фибоначчи +
- Г) Сложение элементов массива

9. Что происходит, если шаблон функции не может быть инстанцирован для определенного типа?

- А) Программа завершится успешно
- Б) Программа выдаст ошибку компиляции +
- В) Компилятор автоматически создаст другой шаблон
- Г) Шаблон будет проигнорирован

10. Какой из следующих вариантов является основной причиной использования шаблонов функций?

- А) Они позволяют писать код, который работает только с одним типом данных
- Б) Они упрощают код и повышают его гибкость, позволяя работать с различными типами данных +
- В) Они увеличивают время компиляции
- Г) Они делают код сложнее для понимания

### **Критерии оценивания тестовых заданий для текущего контроля**

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

## Тест №23

### Тема 23 Указатели. Динамические переменные.

1. Что такое указатель в C++?
  - А) Это переменная, которая хранит значение
  - Б) Это переменная, которая хранит адрес другой переменной +
  - В) Это функция, которая возвращает адрес
  - Г) Это тип данных, который не может быть изменен
  
2. Как правильно объявить указатель на целочисленную переменную?  
`int<i> ptr; +`
  - А) Это неправильный пример
  - Б) Это правильный пример +
  - В) Указатель не должен иметь звездочку
  - Г) Указатель не может ссылаться на целочисленную переменную
  
3. Как можно получить адрес переменной в C++?
  - А) Используя оператор ``
  - Б) Используя оператор `&` +
  - В) Используя оператор `->`
  - Г) Это невозможно
  
4. Какой из следующих вариантов правильно демонстрирует разыменование указателя?  
`int value = 10;`  
`int<i> ptr = &value;`  
`int result = </i>ptr; // result будет равен 10 +`
  - А) Это неправильный пример
  - Б) Это правильный пример +
  - В) Разыменование указателя не работает
  - Г) Указатель не может быть разыменован
  
5. Что произойдет, если вы попытаетесь разыменовать указатель, который не инициализирован?
  - А) Программа завершится успешно
  - Б) Программа выдаст ошибку компиляции
  - В) Произойдет ошибка времени выполнения (сегментация) +
  - Г) Указатель будет автоматически инициализирован нулем
  
6. Как объявить динамическую переменную в C++?
  - А) `int var;`
  - Б) `int var = new int; +`
  - В) `int var;`
  - Г) `int var = new int;`

7. Как освободить память, выделенную для динамической переменной?
- А) delete var; +
  - Б) free(var);
  - В) release(var);
  - Г) clear(var);
8. Что произойдет, если вы забудете освободить память, выделенную для динамической переменной?
- А) Это не повлияет на программу
  - Б) Произойдет утечка памяти +
  - В) Память будет автоматически освобождена
  - Г) Программа завершится с ошибкой
9. Какой оператор используется для удаления динамически выделенной памяти?
- А) free
  - Б) delete +
  - В) dispose
  - Г) clear
10. Какова основная причина использования указателей в C++?
- А) Указатели позволяют передавать большие объемы данных по значению
  - Б) Указатели позволяют работать с динамически выделенной памятью и изменять данные без копирования +
  - В) Указатели упрощают код
  - Г) Указатели делают код более сложным и трудным для понимания

### **Критерии оценивания тестовых заданий для текущего контроля**

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

### **Тест №24**

#### **Тема 24 Динамические массивы**

1. Что такое динамический массив в C++?



- А) Это массив фиксированного размера
- Б) Это массив, размер которого может изменяться во время выполнения программы +
- В) Это массив, который хранит только целые числа
- Г) Это массив, который не может быть инициализирован

2. Как правильно выделить память для динамического массива целых чисел с размером 10?

```
int<i> arr = new int[10]; +
```

- А) Это неправильный пример
- Б) Это правильный пример +
- В) Динамический массив не может иметь размер 10
- Г) Память для массива выделяется с помощью оператора `'malloc'`

3. Как получить доступ к элементу динамического массива?

- А) Используя оператор `'[]'` +
- Б) Используя оператор `'&'`
- В) Используя оператор `'&'`
- Г) Это невозможно

4. Какой оператор используется для освобождения памяти, выделенной для динамического массива?

- А) `delete[]` +
- Б) `free`
- В) `delete`
- Г) `clear`

5. Какой из следующих примеров правильно демонстрирует использование динамического массива?

```
int<i> arr = new int[5];
for (int i = 0; i < 5; ++i) {
 arr[i] = i </i> 10; // Инициализация элементов массива +
}
delete[] arr; // Освобождение памяти +
```

- А) Это неправильный пример
- Б) Это правильный пример +
- В) Раздел `'delete[]'` отсутствует
- Г) Массив не может быть инициализирован в цикле

6. Как изменить размер динамического массива в C++?

- А) Используя оператор `'resize'`
- Б) Это невозможно; нужно создавать новый массив и копировать элементы +

- В) Используя оператор ``new``
- Г) Используя оператор ``adjust``

7. Как можно инициализировать элементы динамического массива при создании?

- А) Использовать цикл и присваивание
- Б) Использовать список инициализации (C++11 и выше) +
- В) Это невозможно
- Г) Использовать оператор ``=``

8. Что произойдет, если вы попытаетесь получить доступ к элементу за пределами динамического массива?

- А) Программа завершится успешно
- Б) Возникнет ошибка времени выполнения (сегментация) +
- В) Будет возвращено значение по умолчанию
- Г) Программа завершится с ошибкой компиляции

10. Как правильно освободить память, выделенную для динамического массива?

- А) `delete var;`
- Б) `delete[] var; +`
- В) `free(var);`
- Г) `clear(var);`

11. Какова основная причина использования динамических массивов в C++?

- А) Динамические массивы позволяют хранить данные фиксированного размера
- Б) Динамические массивы позволяют изменять размер массива во время выполнения программы +
- В) Динамические массивы не требуют освобождения памяти
- Г) Динамические массивы делают код сложнее для понимания

### **Критерии оценивания тестовых заданий для текущего контроля**

Максимальное количество баллов за решение одного теста составляет 5 баллов, которые переводятся в академические оценки следующим образом:

- оценка «отлично» (5 баллов) выставляется при условии правильного ответа студента не менее чем на 85% контрольных заданий
- оценка «хорошо» (4 балла) выставляется при условии правильного ответа студента не менее чем на 70% контрольных заданий
- оценка «удовлетворительно» (3 балла) выставляется при условии правильного ответа студента не менее чем на 51% контрольных заданий
- оценка «неудовлетворительно» (1-2 балла) выставляется при условии правильного ответа студента менее чем на 50% контрольных заданий

## 2.2 Оценочные материалы для проведения промежуточной аттестации

1. Что такое переменная в программировании?  
А) Это функция, которая возвращает значение  
Б) Это область памяти для хранения данных +  
В) Это метод выполнения кода  
Г) Это тип данных
2. Какой из следующих типов данных является целочисленным?  
А) float  
Б) char  
В) int +  
Г) double
3. Какой оператор используется для присваивания значения переменной?  
А) ==  
Б) = +  
В) :=  
Г) =>
4. Какой из следующих вариантов является корректным комментарием в C++?  
А) // Это комментарий +  
Б) /\* Это комментарий  
В) # Это комментарий  
Г) -- Это комментарий
5. Что такое условный оператор?  
А) Это оператор, который всегда выполняется  
Б) Это способ обработки ошибок  
В) Это оператор, который выполняет код в зависимости от условия +  
Г) Это оператор, который завершает программу
6. Какой из следующих операторов используется для сравнения?  
А) =  
Б) == +  
В) :=  
Г) <>
7. Какой из следующих циклов используется для выполнения кода определенное количество раз?  
А) while  
Б) for +  
В) do-while

Г) if

8. Что такое массив?

- А) Это структура данных, которая хранит только одну переменную
- Б) Это коллекция элементов одного типа, доступных по индексу +
- В) Это функция, которая возвращает массив
- Г) Это временная переменная

9. Как объявить массив целых чисел размером 5 в C++?

- А) `int arr[5];` +
- Б) `int arr(5);`
- В) `int arr{5};`
- Г) `int arr = new int[5];`

10. Какой из следующих методов используется для инициализации переменной?

- А) Присваивание +
- Б) Декомпозиция
- В) Инкапсуляция
- Г) Декларация

11. Что такое функция в программировании?

- А) Это блок кода, который выполняет определенную задачу +
- Б) Это переменная, хранящая значения
- В) Это структура данных
- Г) Это тип данных

12. Как вызвать функцию в C++?

- А) `functionName;`
- Б) `call functionName;`
- В) `functionName();` +
- Г) `invoke functionName;`

13. Какой из следующих типов данных является логическим?

- А) `char`
- Б) `int`
- В) `bool` +
- Г) `float`

14. Что такое рекурсия?

- А) Это вызов функции внутри самой себя +
- Б) Это способ сортировки данных
- В) Это метод обработки ошибок
- Г) Это структура данных

15. Какой оператор используется для увеличения значения переменной на 1?
- А) +1
  - Б) ++ +
  - В) += 1
  - Г) +=
16. Какой из следующих операторов используется для логического "И"?
- А) ||
  - Б) && +
  - В) !
  - Г) &
17. Что такое строка в программировании?
- А) Это массив символов +
  - Б) Это числовое значение
  - В) Это функция
  - Г) Это тип данных
18. Какой из следующих методов используется для завершения программы?
- А) exit(); +
  - Б) terminate();
  - В) end();
  - Г) stop();
19. Какой из следующих подходов используется для защиты данных в объектно-ориентированном программировании?
- А) Наследование
  - Б) Инкапсуляция +
  - В) Полиморфизм
  - Г) Абстракция
20. Какой из следующих операторов используется для логического "ИЛИ"?
- А) || +
  - Б) &&
  - В) !
  - Г) |
21. Что такое "поток управления" в программе?
- А) Это порядок выполнения инструкций +
  - Б) Это количество переменных
  - В) Это способ хранения данных
  - Г) Это тип данных
22. Какой из следующих вариантов правильно объявляет функцию, которая возвращает целое число?

int myFunction(); +

- А) Это неправильный пример
- Б) Это правильный пример +
- В) Функция не может возвращать значение
- Г) Функция не может иметь имя

23. Какой из следующих типов циклов выполняется хотя бы один раз?

- А) for
- Б) while
- В) do-while +
- Г) if

24. Установите правильное соответствие между понятиями и их определениями

| Понятия       | Определения                                                                        |
|---------------|------------------------------------------------------------------------------------|
| А) Переменная | 1. Совокупность команд, выполняемых компьютером последовательно.                   |
| Б) Алгоритм   | 2. Место в памяти компьютера, предназначенное для хранения данных.                 |
| В) Цикл       | 3. Конструкция программы, обеспечивающая многократное выполнение блока инструкций. |

Ответ: А-... Б-... В-...

---

Правильный ответ: А-2, Б-1, В-3

25. Вставьте пропущенное слово.

Элементы данных, сгруппированные вместе под одним именем, образуют структуру данных, известную как \_\_\_\_\_.

Ответ: массив

### Критерии оценки для проведения зачета по дисциплине (тестирование)

Шкала оценивания результатов промежуточной аттестации и критерии выставления оценок.

| Оценка                | Уровень сформированности компетенции | Критерии                                        |
|-----------------------|--------------------------------------|-------------------------------------------------|
| «Отлично» («зачтено») | Высокий                              | Выполнено более 80% заданий предложенного теста |
| «Хорошо» («зачтено»)  | Хороший                              | Выполнено 60-80% заданий                        |

|                                         |               |                                                    |
|-----------------------------------------|---------------|----------------------------------------------------|
|                                         |               | предложенного теста                                |
| «Удовлетворительно»<br>(«зачтено»)      | Достаточный   | Выполнено 35-60% заданий<br>предложенного теста    |
| «Неудовлетворительно»<br>(«не зачтено») | Недостаточный | Выполнено менее 35% заданий<br>предложенного теста |

## **Обновление оценочных материалов дисциплины**

**обсуждено и рекомендовано к утверждению** решением кафедры  
наименование кафедры \_\_\_\_\_  
от \_\_\_\_ \_\_\_\_ 20\_\_ г., протокол № \_\_\_\_

**одобрено** Научно-методическим советом \_\_\_\_\_ от \_\_\_\_ \_\_\_\_ 20\_\_ г.,  
протокол № \_\_\_\_