

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет
им. Г. И. Носова»
Многопрофильный колледж



**КОМПЛЕКТ КОНТРОЛЬНО-ОЦЕНОЧНЫХ СРЕДСТВ
ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ
ОП.09 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
ПРОГРАММЫ ПОДГОТОВКИ СПЕЦИАЛИСТОВ СРЕДНЕГО
ЗВЕНА
ПО СПЕЦИАЛЬНОСТИ СПО
09.02.01 Компьютерные системы и комплексы
базовой подготовки**

Магнитогорск, 2017

ОДОБРЕНО

Предметно-цикловой комиссией
«Информатики и вычислительной
техники»

Председатель  И.Г. Зорина
Протокол №7 от 12 марта 2017 г.

Методической комиссией МпК

Протокол № 4 от «23» марта 2017г.

Разработчик:

Преподаватель МпК ФГБОУ ВО «МГТУ им. Г.И. Носова» к.т.н., доцент
В.Д. Тутарова

Комплект контрольно-оценочных средств для текущего контроля и промежуточной аттестации по учебной дисциплине составлен на основе ФГОС СПО по специальности **09.02.01 Компьютерные системы и комплексы** базовой подготовки, утвержденного 28 июля 2014 г. № 849 и рабочей программы учебной дисциплины **Основы алгоритмизации и программирования**

ОБЩИЕ ПОЛОЖЕНИЯ

Учебная дисциплина ОП.09. «Основы алгоритмизации и программирования» относится к общепрофессиональным дисциплинам профессионального цикла.

В результате освоения дисциплины обучающийся **должен уметь**:

- формализовать поставленную задачу;
- выбрать оптимальный метод решения задачи;
- выполнить анализ алгоритма с точки зрения его эффективности;
- применять полученные знания к различным предметным областям;
- составлять и оформлять программы на языках программирования;
- тестировать и отлаживать программы.

В результате освоения дисциплины обучающийся **должен знать**:

- общие принципы построения и использования языков программирования, их классификацию;
- современные интегрированные среды разработки программ;
- процесс создания программ;
- стандарты языков программирования;
- общую характеристику языков ассемблера: назначение, принципы построения и использования.

Содержание дисциплины ориентировано на подготовку студентов к освоению профессиональных модулей ОПОП по специальности и овладению профессиональными компетенциями:

ПК 2.1. Создавать программы на языке ассемблера для микропроцессорных систем.

ПК 2.2. Производить тестирование, определение параметров и отладку микропроцессорных систем.

ПК 3.3. Осуществлять установку и конфигурирование персональных компьютеров и подключение периферийных устройств.

В процессе освоения дисциплины у студентов должны формироваться общие компетенции:

ОК 1. Понимать сущность и социальную значимость своей будущей профессии, проявлять к ней устойчивый интерес.

ОК 2. Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК 3. Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность.

ОК 4. Осуществлять поиск и использование информации,

необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК 5. Использовать информационно-коммуникационные технологии в профессиональной деятельности.

ОК 6. Работать в коллективе и команде, эффективно общаться с коллегами, руководством, потребителями.

ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), результат выполнения заданий.

ОК 8. Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации.

ОК 9. Ориентироваться в условиях частой смены технологий в профессиональной деятельности.

Контроль и оценка результатов освоения учебной дисциплины осуществляется преподавателем в процессе освоения материала: опросы в устной и письменной форме, промежуточное тестирование, оценка самостоятельной работы обучающихся.

В качестве форм и методов текущего контроля используются практические занятия, тестирование, контрольные работы.

Промежуточная аттестация в форме дифференцированного зачета.

. Оценка индивидуальных образовательных достижений по результатам текущего контроля и промежуточной аттестации производится в соответствии с универсальной шкалой:

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

Паспорт оценочных средств

№	Контролируемые разделы (темы) учебной дисциплины	Контролируемые умения, знания	Контролируемые компетенции	Наименование оценочного средства	
				Текущий контроль	Промежуточная аттестация
1.	Введение		ОК.1	Тест входного контроля	Вопросы для дифференцированного зачета
2.	Тема 1.1. Основные понятия алгоритмизации	У1,У4, У5, 31,33,34	ПК.2.1-2.2 ПК.3.3, ОК1-9	Индивидуальное домашнее задание по теме 1.1	
3.	Тема 1.2. Основные алгоритмические конструкции: линейные, разветвляющиеся, циклические.	У1,У4, У5, 31,33,34	ПК.2.1-2.2 ПК.3.3, ОК1-9	Индивидуальное домашнее задание по теме 1.2 Тестирование	
4.	Тема 2.2. Ввод и вывод данных	У1,У4, У5, 31,33,34	ПК.2.1-2.2 ПК.3.3, ОК1-9	Контрольная работа по теме «ввод и вывод данных».	
5.	Тема 2.3. Базовые конструкции языков программирования	У1,У4, У5, 31,33,34	ПК.2.1-2.2 ПК.3.3, ОК1-9	Контрольная работа по теме «оператор условия». Контрольная работа по теме «оператор цикла».	
6.	Тема 2.4. Массивы	У1,У4, У5, 31,33,34	ПК.2.1-2.2 ПК.3.3, ОК1-9	Контрольная работа по теме «массивы». Тестирование	

1. ВХОДНОЙ КОНТРОЛЬ

Входной контроль проводится с целью определения готовности обучающихся к освоению учебной дисциплины, базируется на дисциплинах, предшествующих изучению данной учебной дисциплины:

- Математика
- Информатика

По результатам входного контроля планируется осуществление в дальнейшем дифференцированного и индивидуального подхода к обучающимся.

При низком уровне знаний проводятся корректирующие курсы, дополнительные занятия, консультации.

Примеры заданий входного контроля

1. Расстояние между точками А (-2;3) и В (0;-1) равно:

- а) $2\sqrt{5}$;
- б) 20;
- в) нет правильного ответа.

2. Число с плавающей точкой в компьютере изображаются в виде:

- а) основания системы и мантиссы;
- б) мантиссы и порядка;
- в) определённого количества разрядов.

3. Определите область допустимых значений для функции:

$$f = \frac{\sqrt{x-2}}{(x+2)}$$

4. Запишите жизненную ситуацию в форме:

если <условие выполняется>, то <действие 1>, иначе <действие 2>.

5. Запишите алгоритм заполнения 7-ми литрового ведра литровыми банками в форме:

- а) пока <выполняется условие> выполнить <действие>
- б) повторить <действие> до тех пор, пока <условие ложно>
- в) для i от 1 до $\langle r \rangle$ выполнять <действие>

6. Ответьте на вопрос. Заданы 4 переменные x , y , z , k . Каждая переменная может принимать только два значения: **истина**, **ложь**. Сколько всевозможных комбинаций значений имеют эти переменные?

- а) 8
- б) 16
- в) 24

Эталоны ответов: 1. В; 2. Б; 3. [2; +∞); 6. Б;

Критерии оценки

За каждый правильный ответ – 1 балл.

За неправильный ответ – 0 баллов.

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

2. ТЕКУЩИЙ КОНТРОЛЬ

Текущий контроль успеваемости осуществляется в ходе повседневной учебной работы по курсу дисциплины по инициативе преподавателя. Данный вид контроля стимулирует стремление к систематической самостоятельной работе по изучению учебной дисциплины, овладению профессиональными и общими компетенциями, позволяет отслеживать положительные/отрицательные результаты и планировать предупреждающие/корректирующие мероприятия.

Формы текущего контроля

1. Устный опрос
2. Самостоятельная работа
3. Тестирование
4. Контрольная работа
5. Контроль самостоятельной работы студентов
6. Контроль выполнения практических работ

2.1 Тестирование

1. Алгоритм – это
 1. понятное и точное предписание исполнителю совершить последовательность действий, направленных на решение поставленной задачи
 2. некоторые истинные высказывания, которые должны быть направлены на достижение поставленной цели
 3. отражение предметного мира с помощью знаков и сигналов, предназначенное для конкретного исполнителя
 4. инструкция по технике безопасности
2. Свойствами алгоритма являются
 1. новизна, понятность, массовость, дискретность, результативность
 2. дискретность, результативность, детерминированность, массовость, понятность
 3. массовость, понятность, условность, четкость, однозначность
 4. четкость, однозначность, массовость, дискретность, результативность
3. Расчлененность алгоритма на отдельные шаги, возможность выполнения которых исполнителем не вызывает сомнений, отражена в свойстве алгоритма, которое называется
 1. дискретностью
 2. однозначностью
 3. результативностью

4. понятностью
4. Основное свойство алгоритма, характерное только для решения задач на ЭВМ, это
 1. дискретность
 2. понятность
 3. массовость
 4. точность
5. Свойство алгоритма – дискретность – обозначает
 1. разбиение алгоритма на конечное число простых шагов
 2. что команды должны следовать последовательно друг за другом
 3. что каждая команда должна быть описана в расчете на конкретного исполнителя
 4. нет верного ответа
6. Графическое задание алгоритма – это
 1. представление алгоритма в форме таблиц и расчетных формул
 2. система обозначения правил для единообразной и точной записи алгоритмов и их исполнения
 3. способ представления алгоритма с помощью геометрических фигур
 4. схематичное изображение в произвольной форме
7. Выбор метода решения должен предшествовать этапу
 1. тестирования и отладки
 2. разработки алгоритма
 3. построения математической модели
 4. анализа и уточнения результатов
8. Запись алгоритма в виде последовательности команд компьютера называется
 1. вербальной
 2. графической
 3. программной
 4. словесной
9. На этапе тестирования и отладки происходит
 1. получение результата
 2. перевод алгоритма на алгоритмический язык
 3. представление задачи в виде последовательности математических формул
 4. обнаружение и исправление синтаксических ошибок, и доведение программы до рабочего состояния
10. Компьютер может отследить
 1. синтаксические ошибки
 2. логические ошибки
 3. фактические ошибки в формулах

любые ошибки

11. Завершимость алгоритма за конечное число шагов отражено в свойстве, которое называется

1. однозначностью
2. результативностью
3. понятностью
4. дискретностью

12. Прикладные программы

1. программы, предназначенные для решения конкретных задач
2. управляют работой аппаратных средств и обеспечивают услуги нас и наши прикладные комплексы
3. игры, драйверы, трансляторы и т.д.
4. программы, которые хранятся на различного типа дисках

13. Программа

1. это набор инструкций на машинном языке, который хранится в виде файла на магнитном диске и по вашей команде загружается в компьютер для выполнения
2. это игры, предназначенные для использования на ЭВМ
3. это набор инструкций, предназначенный для запуска компьютера
4. это набор инструкций, предназначенных для работы компьютера

Критерии оценки

За каждый правильный ответ – 1 балл.

За неправильный ответ – 0 баллов.

Процент результативности (правильных ответов)	Качественная оценка индивидуальных образовательных достижений	
	балл (отметка)	вербальный аналог
90 ÷ 100	5	отлично
80 ÷ 89	4	хорошо
70 ÷ 79	3	удовлетворительно
менее 70	2	не удовлетворительно

2.2 Контрольные работы

Спецификация

Контрольные работы проводятся после изучения соответствующего раздела.

Тема 2.3. Ввод и вывод данных

1. Определить тип результата следующих выражений:
 - а. $I * I + J * J + 2 * K / 2$
 - б. $x < y$
 если известно, что I, J, K - int; x, y - float?
2. Какие из приведенных ниже операторов присвоения являются правильными?
 - а. $x := y + \sin(\sin(z))$
 - б. $A := (x < y) \text{ or } B \text{ and } (I < > k)$
 - в. $x := I + J - B$
 - г. $I := I + K / J$
 если известно, что I, J, K - int; x, y, z, A - float; D : bool?
3. Организовать ввод и вывод данных заданных типов, снабдив распечатки соответствующими заголовками

S=36	M=21425	Y=21.345	S1='*'
	S2='V'		

 Дать протокол программы.

Тема 2.3. Базовые конструкции языков программирования

1. Услуги телефонной сети оплачиваются по следующему правилу: разговоры до A минут в месяц — B руб., а разговоры сверх установленной нормы оплачиваются из расчета C руб. за минуту. Написать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.
2. Грузовой автомобиль выехал из одного города в другой со скоростью v_1 км/ч. Через t ч в этом же направлении выехал легковой автомобиль со скоростью v_2 км/ч. Составить программу, определяющую, догонит ли легковой автомобиль грузовой через t_1 ч после своего выезда.

Операторы цикла

1. Даны числовой ряд и некоторое число ε . Найти сумму членов ряда, модуль которых больше или равен заданному ε . Общий член ряда имеет вид:

$$a_n = \frac{(-1)^{n-1}}{n^n}$$

2. Найти наименьший номер члена последовательности, для которого выполняется условие $|a_n - a_{n-1}| < \varepsilon$. Вывести этот номер и все элементы a_i , где $i=1, 2, \dots, n$.

$$a_n = 2 + \frac{1}{a_{n-1}}, \text{ где } a_1 = 2$$

Массивы

1. В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах.

Составить алгоритм и написать программу.

2. Даны действительные числа a_1, a_2, \dots, a_n . Среди них есть положительные и отрицательные. Заменить нулями те числа, величина которых по модулю больше максимального числа:

$$(|a_i| > \max\{a_1, a_2, \dots, a_n\})$$

3. ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Промежуточная аттестация обучающихся по учебной дисциплине, осуществляется по завершении изучения данной дисциплины и позволяет определить качество и уровень ее освоения. Предметом оценки освоения учебной дисциплины являются умения и знания.

3.1 Вопросы дифференцированного зачета

1. Структура программы на языке C++.
2. Основные типы переменных в C. Правила определения переменных и типов. Инициализация данных.
3. Время жизни и область видимости программных объектов. Инициализация глобальных и локальных переменных
4. Операторы языка Си. Оператор выражение, составной оператор, операторы условного перехода
5. Оператор цикла **for**
6. Операторы цикла **while** и **do ... while**
7. Форматированный ввод-вывод.
8. Указатели. Типизированные и нетипизированные.
9. Работа с файлами. Файловая переменная, открытие, закрытие файла.
10. Чтение и запись в файлы. Функция **eof()**
11. Определение и вызов функций. Фактические и формальные параметры.
12. Определение и вызов функций. Передача массивов и указателей на функции.
13. Бинарный поиск в упорядоченном массиве.
14. Сортировка массива на примере одного из алгоритмов.
15. Связанные списки: описание структуры, добавление и удаление элементов в односвязный линейный список.
16. Виды линейных списков: стек, очередь, дек.
17. Описание структуры на языке C++. Определение переменных структурного типа. Способы доступа к элементам структур.
18. Описание объединения на языке C++. Определение переменных типа «объединение». Способы доступа к элементам объединений.
19. Сущность методологии объектно-ориентированного программирования. Инкапсуляция. Наследование. Полиморфизм.
20. Определение класса на языке C++. Ограничение доступа к членам класса.
21. Дружественные функции класса.
22. Определение конструктора. Форматы вызова конструктора. Перегрузка конструкторов.

23. Наследование. Определение базового и производного классов.
24. Полиморфизм. Виртуальные функции.
25. Динамическое распределение памяти под объекты. Операторы new и delete.
26. Шаблоны классов.
27. Шаблоны функций.
28. Понятие потока в языке C++. Стандартные потоки.

Раздел 2. Структурное программирование

Теоретические вопросы

1. Когда целесообразно использовать консольные приложения?
2. Какой интерфейс пользователя имеют консольные приложения?
3. Что включает шаблон консольного приложения?
4. В чем назначение операторов программы?
5. Какую конструкцию программы называют блоком?
6. В чем назначение переменных программы?
7. Что такое именованная константа?
8. В каких случаях целесообразно объявлять переменные с начальными значениями?
9. Для чего предназначены директивы `#include`?
10. Какие виды ошибок бывают в программе?
11. Такие ошибки препятствуют созданию исполняемого файла?
12. Какими способами можно запустить консольное приложение?
13. Что понимают под отладкой программы?
14. Что такое тестирование?
15. Какие средства есть в Visual C++ для поиска смысловых ошибок в программе?
16. Что называют точкой безусловного останова?
17. Как приостановить на заданной строке программы её выполнение в отладочном ре-жиме?
18. Какие бывают виды трассировки?
19. Какими способами можно наблюдать за значениями переменных в процессе отладки программы?
20. Как создать панель инструментов отладки или изменить набор её инструментов?
21. Для чего предназначены комментарии в программе?
22. Какие есть виды комментариев?
23. Как можно быстро превратить часть строк программы в комментарий?
24. Как можно создать отладочный вариант программы, используя директивы условной компиляции, чтобы после отладки можно было быстро восстановить рабочий вариант программы?

3.2 Типовые задания

Раздел 2 Структурное программирование

Вопрос 1

Укажите, какое из нижеследующих утверждений ложное.

- 1: Комментарии при выполнении программы вызывают печать компьютером на экране текста после символов //.
- 2: Если вывод осуществляется в cout, то esc-последовательность \n вызывает перемещение курсора к началу следующей строки на экране.
- 3: Все переменные должны быть объявлены до того, как они используются.
- 4: Все высказывания истинные

Вопрос 2

Укажите, какое из нижеследующих утверждений ложное.

- 1: Всем переменным, когда они объявляются, должен быть присвоен тип.
- 2: C++ рассматривает переменные number и NuMbEr как одинаковые.
- 3: Объявления в теле функции C++ могут появляться почти везде.
- 4: Все высказывания истинные

Вопрос 3

Укажите, какое из нижеследующих утверждений ложное.

- 1: Операция взятия по модулю (%) может применяться только к целым числам.
- 2: Все арифметические операции *, /, %, + и - имеют одинаковый уровень приоритета.
- 3: Пустые скобки, следующие за именем функции в прототипе, указывают, что функции для выполнения ее задачи не требуется никаких параметров.
- 4: Все высказывания истинные

Вопрос 4

Укажите, какое из нижеследующих утверждений ложное.

- 1: Тело каждой функции ограничивается левой и правой фигурными скобками ({ и }).
- 2: Программа на C++, которая выводит три строки, должна содержать три оператора вывода, использующих cout.
- 3: Переменная типа int может присвоить значение дробного типа.
- 4: Все высказывания истинные

Вопрос 5

Укажите, какое из нижеследующих утверждений истинное.

- 1: В соответствии с соглашением, имена функций начинаются с прописной буквы, и все последующие слова в имени также начинаются с прописной буквы.
- 2: Переменные, объявленные в теле некоторой элемент - функции, называются элементами данных и могут использоваться в функции main().
- 3: Для исполнения программы можно использовать любой файл исходного кода, содержащий int main().
- 4: Указатель, объявленный как void, может быть разыменован.

Вопрос 6

Укажите, какое из нижеследующих утверждений истинное.

- 1: Типы аргументов в вызове функции должны быть согласованы с типами соответствующих параметров в списке параметров прототипа функции.
- 2: В операторе выбора switch должна быть метка default.
- 3: В операторе выбора switch в разделе default требуется оператор break.
- 4: Все высказывания истинные

Вопрос 7

Укажите, какое из нижеследующих утверждений истинное.

- 1: Выражение, содержащее операцию ||, истинно, если истинны оба операнда этой операции.
- 2: Массив может хранить много различных типов данных.
- 3: Операция взятия адреса может быть применима только к константам, к выражениям, не дающим в качестве результата ссылки, и к переменным, объявленным с классом памяти register.
- 4: Все высказывания ложные

Вопрос 8

Укажите, какое из нижеследующих утверждений истинное.

- 1: Если количество начальных значений в списке инициализации меньше, чем количество элементов массива, оставшиеся элементы автоматически получают в качестве начальных значений последние значения из списка инициализации.
- 2: Если список инициализации содержит начальных значений больше, чем элементов массива, то это - ошибка.
- 3: Отдельный элемент массива, который передается функции и модифицируется в этой функции, будет содержать модифицированное значение после завершения выполнения вызываемой функции.
- 4: Все высказывания ложные

Вопрос 9

Укажите, какое из нижеследующих утверждений истинное.

- 1: Когда исполнение программы приостанавливается в контрольной точке, следующим оператором, который будет исполняться, является оператор после контрольной точки.
- 2: В операторе условия необходимо использования оператора Then.
- 3: Во время отладки команда Step Out исполняет все оставшиеся операторы текущей функции и возвращает управление в точку, где была вызвана функция.
- 4: Все высказывания истинные

Вопрос 10

Укажите, какое из нижеследующих утверждений ложно.

- 1: Указатели разных типов нельзя присваивать друг другу без операции приведения типа.
- 2: Выражение $(x > y \ \&\& \ a < b)$ истинно, если $x > y$ или $a < b$.
- 3: Все высказывания ложные
- 4: Индексы массива обычно должны иметь тип float.

Вопрос 11

Выполнение каждой программы на C++ начинается с функции_____.

- 1: void
- 2: include
- 3: main
- 4: using

Вопрос 12

Каждый оператор заканчивается_____.

- 1: endl
- 2: return
- 3: точкой с запятой
- 4: управляющей последовательностью

Вопрос 13

Оператор_____ используется для принятия решений

- 1: cout
- 2: if
- 3: while
- 4: for

Вопрос 14

Оператор выбора _____ используется для выполнения одного действия, если его условие истинно, и другого действия, если условие ложно

- 1: If
- 2: If...then
- 3: If...then...else
- 4: If...else

Вопрос 15

Повторение набора инструкций заданное число раз называется _____ повторений

- 1: управляемым счетчиком
- 2: числом
- 3: заданием

Вопрос 16

Когда заранее не известно, сколько раз должна исполняться группа операторов, для прерывания повторения можно использовать _____ значение

- 1: контрольное
- 2: сигнальное
- 3: фиктивное
- 4: все ответы верны

Вопрос 17

Списки и таблицы значений хранятся в _____

- 1: массиве
- 2: указателе
- 3: списке
- 4: стандартной библиотеке

Вопрос 18

Элементы массива связаны тем, что они имеют одни и те же _____

- 1: имя
- 2: тип
- 3: значения
- 4: размерность

Вопрос 19

Число, используемое для обращения к отдельному элементу массива называется _____

- 1: тип

- 2: индекс
- 3: значение
- 4: позиция

Вопрос 20

Для объявления размера массива должна использоваться _____, потому что она делает программу более масштабируемой

- 1: переменная
- 2: константа
- 3: именованная константа
- 4: символ

Вопрос 21

Процесс упорядоченного размещения элементов в массиве называется _____

- 1: сортировка
- 2: поиск
- 3: проверка
- 4: изменение

Вопрос 22

Процесс определения значения ключа, содержащегося в массиве, называется _____

- 1: сортировка
- 2: поиск
- 3: проверка
- 4: изменение

Вопрос 23

Указатель - это переменная, которая содержит в качестве своего значения _____ другой переменной

- 1: индекс
- 2: адрес
- 3: код
- 4: ссылку

Вопрос 24

Для инициализации указателя можно использовать

- 1: null
- 2: 0
- 3: адрес
- 4: все перечисленное

Вопрос 25

Целое, которое может быть присвоено указателю, является

- 1: 0
- 2: положительным
- 3: отрицательным
- 4: все перечисленное

Вопрос 26

Что такое идентификаторы в языке Си++ ?

- 1: это последовательность знаков, начинающаяся с буквы или знака подчеркивания
- 2: это последовательность знаков
- 3: это последовательность знаков, начинающаяся с буквы
- 4: это последовательность знаков, начинающаяся со знака подчеркивания

Вопрос 27

В идентификаторах можно использовать

- 1: заглавные и строчные латинские буквы, цифры и знак подчеркивания
- 2: заглавные и строчные латинские буквы, цифры и знак подчеркивания
- 3: заглавные и строчные латинские буквы
- 4: заглавные и строчные латинские буквы, цифры
- 5: заглавные и строчные латинские буквы, знак подчеркивания

Вопрос 28

Что является выражениями в языке Си++ ?

- 1: это переменные, функции и константы объединенные знаками операций
- 2: это переменные объединенные знаками операций
- 3: это функции объединенные знаками операций
- 4: это константы объединенные знаками операций

Вопрос 29

Какая из операций не относится к операции сравнения?

- 1: (=)
- 2: (>)
- 3: (<)
- 4: (==)
- 5: (!=)

Вопрос 30

В результате успешной компиляции текста программы на C++ с каким расширением будет получен файл?

- 1: .obj
- 2: .exe
- 3: .cpp
- 4: .h

Вопрос 31

Если не будет указан базовый тип, то какой тип будет подразумеваться по умолчанию

- 1: int
- 2: char
- 3: short
- 4: long

Вопрос 32

К чему применяется операция инкремента?

- 1: к переменным
- 2: к константам
- 3: к выражениям
- 4: к строкам

Вопрос 33

К каким числам применима операция %?

- 1: к целым
- 2: к вещественным
- 3: и к целым и к вещественным

Вопрос 34

Какой операции нет в C++?

- 1: последовательной
- 2: унарной
- 3: бинарной
- 4: тернарная

Вопрос 35

Из чего состоит оператор объявления имени?

- 1: из названия типа и объявляемого имени
- 2: из названия типа
- 3: из объявляемого имени

Вопрос 36

Оператор while соответствует оператору for:

- 1: for (; условие ;)
оператор
- 2: for (условие ; ;)
оператор
- 3: for (; ; условие)
оператор

Вопрос 37

В языке Си++ указатель - это:

- 1: специальный значок, показывающий, что это динамическая переменная
- 2: символическое представление адреса ячейки памяти
- 3: символ, указывающий на что-либо
- 4: метка

Вопрос 38

Комментарии заключаются в скобки:

- 1: { }
- 2: /* */
- 3: []
- 4: /% %/

Вопрос 39

Идентификатор - это:

- 1: последовательность латинских букв, цифр и символа «_», начинающаяся с буквы или символа «_»
- 2: неизменяемые объекты языка (константы)
- 3: последовательность латинских и русских букв
- 4: способ кодирования, допустимые преобразования над значением данной переменной

Вопрос 40

Фактический адрес в указателях - это:

- 1: строка
- 2: указатель
- 3: число
- 4: буква

Вопрос 41

Составной оператор - это:

- 1: последовательность операторов, заключенная в фигурные скобки { }
- 2: последовательность операторов, заключенная квадратные скобки []

3: последовательность операторов, заключенная в операторные скобки
begin ... end

4: последовательность операторов, заключенная в круглые скобки ()

Вопрос 42

Логическое «не равно» обозначается:

1: <>

2: ||

3: !

4: !=

Вопрос 43

Логическое «и» обозначается:

1: =

2: ||

3: &

4: &&

Вопрос 44

Логическое «не» обозначается:

1: !

2: !!

3: ||

4: not

Вопрос 45

Битовая операция инверсии битов обозначается:

1: \~

2: ~

3: >>

4: <<

Вопрос 46

Битовая операция исключающего «или» обозначается:

1: \~

2: ~

3: ||

4: &&

Вопрос 47

Операция битового «и» обозначается:

1: \~

- 2: ~
- 3: ||
- 4: &

Вопрос 48

Операция битового «или» обозначается:

- 1: \~
- 2: -
- 3: ||
- 4: &

Вопрос 49

Выражения - это:

- 1: конструкции, включающие константы (литералы), переменные, знаки операций, скобки для управления порядком выполнения операций, обращения к функциям
- 2: основные строительные блоки программы; в языке Си++ указанием на наличие выражения служит символ «точка с запятой», стоящий в конце него
- 3: набор символов и операций
- 4: операторы, выполняющие определенные действия с переменными

Вопрос 50

Оператор-переключатель - это:

- 1: оператор для выбора одного из нескольких вариантов (SWITCH)
- 2: строка с меткой DEFAULT
- 3: CASE
- 4: BREAK

Вопрос 51

Оператор цикла DO/WHILE является:

- 1: конструкцией цикла с предусловием
- 2: конструкцией цикла с постусловием
- 3: конструкцией цикла с выбором варианта
- 4: конструкцией цикла с перебором значений параметра

Вопрос 52

Формальный аргумент - это:

- 1: конкретное значение, присвоенное этой переменной вызывающей программой
- 2: переменная и вызываемой программе
- 3: строка, которая пишется в скобках функции

4: строка, которая пишется в скобках процедуры

Вопрос 53

Фактический аргумент - это:

- 1: конкретное значение, присвоенное этой переменной вызывающей программой
- 2: переменная в вызываемой программе
- 3: строка, которая пишется в скобках функции
- 4: строка, которая пишется в скобках процедуры

Вопрос 54

Писать `# include <stdio.h>` нужно для:

- 1: подключения файла, содержащего макроопределения и объявления данных, необходимых для работы функций из стандартной библиотеки ввода-вывода
- 2: позволяет дать в программе макроопределения (или задать макросы)
- 3: переопределения не только константы, но и целых программных конструкций
- 4: замены каждого параметра в строке лексем на соответствующий аргумент макровывода

Вопрос 55

Точка с запятой является:

- 1: разделителем операторов
- 2: частью оператора
- 3: ключевым знаком языка Си
- 4: спецсимвол

Вопрос 56

В языке Си++ тело функции ограничено операторными скобками:

- 1: `begin end`
- 2: `start finish`
- 3: `[]`
- 4: `{ }`

Вопрос 57

Оператор `INT` в Си++ применяется для:

- 1: переопределения диапазона целых чисел
- 2: преобразования переменной к целому типу
- 3: описания переменных целого типа
- 4: прибавления единицы к коду символа

Вопрос 58

Наличие нуль-символа (`\0`) означает, что:

- 1: количество ячеек массива должно быть, по крайней мере, на одну больше, чем число символов, которые необходимо размещать в памяти
- 2: логическим значением переменной является «ложь»
- 3: количество ячеек массива должно быть на одну меньше, чем число символов, которые необходимо размещать в памяти
- 4: логическим значением переменной является «истина»

Вопрос 59

Пример: `val = *ptr`; операция косвенной адресации `*` производит:

- 1: получение адреса
- 2: перенаправление адреса переменной `val` к переменной `ptr`
- 3: определение значения, на которое указывает `ptr`
- 4: определение значения, на которое указывает `val`

Вопрос 60

Если в цикле задано два разных условия выхода, то используется оператор:

- 1: `CONTINUE`
- 2: `BREAK`
- 3: `GOTO`
- 4: `NEXT`

Вопрос 61

Если в выражениях встречаются операнды различных типов, то они преобразуются к общему типу в соответствии с определенными правилами. Если один из операндов имеет тип `char`, то:

- 1: другие также преобразуются к типу `char` и результат имеет тип `Char`
- 2: другие преобразуются к типу `int` и результат имеет тип `int`
- 3: во время операции присваивания значение правой части преобразуется к типу левой части, который и становится типом результата
- 4: остается как есть и результат будет `Char`

Вопрос 62

Метки в операторе `Switch` должны быть:

- 1: указателями
- 2: переменной
- 3: константой
- 4: типа `Char`

Вопрос 63

Используя форму обращения Function1 (&x), получаем:

- 1: передачу в функцию значения переменной x
- 2: передачу адреса переменной x
- 3: использование глобальной переменной
- 4: использование класса памяти x

Вопрос 64

Тип функции определяется:

- 1: типом ее аргументов
- 2: использованием в программе
- 3: типом ее описания
- 4: типом возвращаемого ею значения

Вопрос 65

Каков будет результат выполнения операторов:

```
nrs = 22;  
ptr = &nrs;  
val = *ptr;
```

- 1: присваивание значения 22 переменной ptr
- 2: &nrs дает адрес переменной val
- 3: &nrs дает адрес переменной ptr
- 4: присваивание значения 22 переменной val

Вопрос 66

Каков будет результат выполнения операторов:

```
int i,j,s;  
i=j=2; /* i и j получают значение 2 */  
s=(i++)+(++j);
```

- 1: i = 3, j = 2, s = 5
- 2: i = 3, j = 3, s = 6
- 3: i = 3, j = 3, s = 5
- 4: i = 2, j = 3, s = 5

Вопрос 67

Каков будет результат выполнения операторов:

```
int x,y,a;  
x=5;  
y=x*2+7;  
a=y/4;
```

- 1: x = 5, y = 17, a = 4,25
- 2: x = 5, y = 17, a = 4
- 3: x = 5, y = 10, a = 2,25

4: $x = 5, y = 32, a = 8$

Вопрос 68

Каков будет результат выполнения операторов:

$a = (y = (x = 5) * 2 + 7) / 4$

1: $a = 4,25$

2: $a = 4$

3: $a = 2,25$

4: error

Вопрос 69

Каков будет результат выполнения операторов:

`int x,y;`

`x=y=5;`

`x+=2;`

`y-=3;`

`x*=y;`

`x/=++y;`

1: $y = 3, x = 4$

2: $y = 4, x = 12$

3: $y = 12, x = 12/3$

4: $y = 3, x = 14$

Вопрос 70

Каков будет результат выполнения операторов:

`int a,b`

`a = 4;`

`b = 7;`

`m = (a > b) ? a : b;`

1: $m = 4$

2: $m = 11$

3: $m = 3$

4: $m = 7$

Вопрос 71

Каков будет результат выполнения операторов:

`int x,y`

`y = -4;`

`x = (y < 0) ? -y : y;`

1: $x = 4$

2: $x = -4$

3: $x = 0$

4: x: = 8

Вопрос 72

Определите, что будет выведено на экран в результате выполнения приведенных фрагментов кода. Объясните работу этих конструкций пошагово.

а) `int C[3]={1,2,3};
for(int i=0; i<3; i++)
C[0] += C[i];
cout << C[0];`

б) `int C[3]={1,2,3};
for(int i=0; i<3; i++)
C[i] += C[2-i];
cout << C[0]+C[1]+C[2];`

в) `int I[3]={3,2,1};
for(int j=0; j<3; j+=2)
cout << I[j];`

г) `int S[3]={1}, f=1;
for(int i=0; i<3; i++)
f += S[i];
cout << f;`

Вопрос 73

Определите, что будет выведено на экран в результате выполнения приведенных фрагментов кода. Объясните работу этих конструкций пошагово.

а) `int A[2][2]={{1,2},{3,4}};
for(int i=0; i<2; i++)
for(int j=0; j<2; j++)
A[i][j] += 1;
cout << A[0][1]+A[1][0];`

б) `int Z[4]={1,0,1,0};
for(int i=0; i<4; i++)
if(Z[i]>0)
cout << Z[i];`

в) `int X[2][2]={{0,1},{1,0}};
int p = 0;
for(int i=0; i<2; i++)
p += X[0][i];
cout << p;`

г) `int Y[2][2]={{0,1},{1,0}};
int r = 0;
for(int i=0; i<2; i++)
r += Y[i][0];
cout << r;`

Вопрос 74

Определите, что будет выведено на экран в результате выполнения приведенных фрагментов кода. Объясните работу этих конструкций пошагово.

```
a) if(a > b)
    a = b + 1;
    else
    b = a - 1;
    cout<<a<<b<<c;
```

```
б) if(5 > b)
    {
    a = b + 1;
    b = a - 1;
    }
    else
    cout<<a<<b<<c;
```

```
в) if(a < 0)
    a = 10;
    cout<<a<<b<<c;
```

```
г) if(a > 0)
    cout<<"Ok!";
    else
    b = a;
    cout<<a<<b<<c;
```

```
д) if((a < b)||(b > c))
    a = b;
    else
    b = a;
    cout<<a<<b<<c;
```

```
е) if((a < b)&&(b > c))
    a = b;
    else
    b = a;
    cout<<a<<b<<c;
```

Вопрос 75

Определите, что будет выведено на экран в результате выполнения приведенных ниже фрагментов. Предполагается, что переменные i, j, k объявлены как целочисленные (int) и инициализированы значениями 1, 5 и 10, соответственно.

```
a) for(i=0; i<3; i++)
    k = k+1;
    cout<<i<<j<<k<<endl;
```

```
б) for(i=0; i<3; i++)
    {
    k++;
    cout<<k;
    }
```

```
в) for(i=5; i>=3; i--)
    cout<<i<<endl;
```

```
г) for(i=5; i>0; i=i-2)
    cout<<i<<endl;
```

```
д) for(i=0; i<10; i++)
    if(i<j)
    cout<<i;
```

```
е) for(i=0; i<10; i+=2)
    if(i<j)
    cout<<j;
```

```
ж) for(i=0; i<3; i++)
{
    k = k-5;
    if(i<k)
        printf("$$$");
    else
        printf("###");
}
```

```
з) for(i=0; i<3; i++)
{
    for(j=0; j<3; j++)
    {
        if(i<j)
            printf("$$$");
        else
            printf("###");
    }
}
```

Вопрос 76

Найдите синтаксические и логические ошибки в приведенных конструкциях. Считайте, что переменные i, j объявлены ранее как целые (int), переменные x, y – как вещественные (float).

```
а) for(i=0, i<5, i++)
    j++;
```

```
б) for(i=0; i<5; i++)
    j++;
```

```
в) for(int k=0; k<=j; k=k+2)
    cout<<"k!";
else
    cout<<"k?";
```

```
г) for(int i=10; i<=1; i--)
{
    j *= 2;
    cout<<j;
}
```

```
д) for(j=0; j<2; j+=0.1)
{
    x = sin(j*0.1);
    y = exp(-2*j);
}
```

```
е) {for(j=0; j<2; j++)
    x = j*0.1;
    cout<<x;
}
```

Вопрос 77

Найдите синтаксические и логические ошибки в приведенных ниже конструкциях.

```
а) int i, j;
float D[i][j];
```

```
б) int a;
int a[3] = a;
```

```
в) const double N = 2;
double Matrix[N];
```

```
г) const int S;
S = 5;
float G[S];
```

```
д) int N = 4;
double Matrix[N];
```

```
е) int Q[3,3];
Q[0,0] = 1.;
```

```
ж) int L[2][2][2];
L[2][2][2] = 0;
```

```
з) float A[3];
for(int i=0; i<=3; i++)
A[i] = sin(i/10.);
```

```
и) const int r = 2;
int L[r] = 0;
```

```
к) char str[4] = { 1.2, 3.5,
8.0,-1.3};
```

Вопрос 78

Определите, что будет выведено на экран в результате выполнения приведенных ниже фрагментов. Объясните работу этих конструкций пошагово.

```
а) int A[5]={1,2,3,4,5};
cout << "A[0]"
<< A[0];
```

```
б) int b[3]={9,8,7};
cout << b[0] << endl
<< b[1] << endl
<< b[2];
```

```
в) int C[5];
for(int i=0; i<5; i++)
C[i] = 2*i;
cout << C[3];
```

```
г) int DDD[5];
for(int k=0; k<5; k++)
{
DDD[k] = 0.;
cout << "!" ;
}
```

```
д) int E[3]={0,1,2};
E[0]++;
int r=E[0]+E[1]+E[2];
cout << r;
```

```
е) int f[3]={0,1,2};
for(int k=2; k>=0; k--)
cout << f[k];
```

```
ж) int g[2][2] = {{1,2},
{3,4}};
for(int i=0; i<2; i++)
cout << g[i][i];
```

```
з) int H[2][2] = {{5,6},
{7,8}};
for(int i=0; i<2; i++)
cout << H[i][1-i];
```

Вопрос 79

Определите, что будет выведено на экран в результате выполнения приведенных фрагментов кода. Объясните работу этих конструкций пошагово.

```
а) int C[3]={1,2,3};
for(int i=0; i<3; i++)
C[0] += C[i];
cout << C[0];
```

```
б) int C[3]={1,2,3};
for(int i=0; i<3; i++)
C[i] += C[2-i];
cout << C[0]+C[1]+C[2];
```



```
в) int I[3]={3,2,1};
for(int j=0; j<3; j+=2)
cout << I[j];
```

```
г) int S[3]={1}, f=1;
for(int i=0; i<3; i++)
f += S[i];
cout << f;
```

```
д) int A[2][2]={{1,2},{3,4}};
for(int i=0; i<2; i++)
for(int j=0; j<2; j++)
A[i][j] += 1;
cout << A[0][1]+A[1][0];
```

```
е) int Z[4]={1,0,1,0};
for(int i=0; i<4; i++)
if(Z[i]>0)
cout << Z[i];
```

```
ж) int X[2][2]={{0,1},{1,0}};
int p = 0;
for(int i=0; i<2; i++)
p += X[0][i];
cout << p;
```

```
з) int Y[2][2]={{0,1},{1,0}};
int r = 0;
for(int i=0; i<2; i++)
r += Y[i][0];
cout << r;
```

Вопрос 80

Рассмотрите следующие определения функций. Дайте их краткое описание (имя, тип результата, количество и тип аргументов). Одной фразой сформулируйте цель, которую реализует каждая из этих функций (например, «вычисляет сумму двух аргументов», «выводит на экран целые числа от 0 до 100» и т.д.).

```
а) int P(int x1, int x2)
{
    int x3 = x2*x1;
    return x3;
}
```

```
б) int S(int x, int y)
{
    if(x>y) return y;
    else return x;
}
```

```
в) void myprint(int x)
{
    cout << "x=" << x;
    return;
}
```

```
г) void say_yes(void)
{
    cout << "Yes!";
    return;
}
```

```

д) void repeat(char symbol)
{
    for(int i=0; i<20; i++)
        cout << symbol;
    return;
}

```

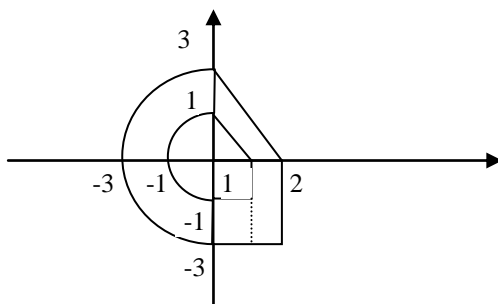
```

е) void minpow(float x, int k)
{
    float y = pow(x, k);
    cout << y;
    return;
}

```

3.2 Типовые задания

1. Определить, попадет ли точка с координатами (x, y) в указанную область.



2. Дана последовательность целых чисел из n элементов. Найти:
 - - среднее арифметическое;
 - - (максимальное значение);
 - - количество отрицательных элементов;
 - - номер минимального элемента;
 - - количество четных чисел;
 - - минимальный из четных элементов этой последовательности.
3. Найти сумму цифр числа n .
4. Дан массив целых чисел. Перевернуть массив.
5. Дан массив целых чисел. Поменять местами пары элементов в массиве: 1 и 2, 3 и 4, 5 и 6 и т. д.
6. Дан массив целых чисел. Упорядочить в порядке убывания.
7. Найти первое вхождение элемента K в массив целых чисел.
8. Удалить из динамической матрицы строку с номером K .
9. Сформировать динамический массив строк. Удалить из него строку с заданным номером.
10. Заданы координаты сторон треугольника. Если такой треугольник существует, то найти его площадь. Решить задачу с использованием функций.

11. Дан массив `int a[100]`. Подсчитать все четные элементы.
12. Дан массив `int *a`. Удалить из массива все элементы, совпадающие с первым элементом, используя динамическое выделение памяти.
13. Сформировать одномерный массив состоящий из `n` случайных значений в диапазоне от 1 до 10. Записать массив в текстовый файл.
14. Сформировать одномерный массив состоящий из `n` случайных значений в диапазоне от 1 до 10. Записать массив в бинарный файл.
15. Добавить в бинарный файл, в который записаны элементы типа `struct Student`
`{char name[20];int age;};`