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
ОСНОВЫ ПРОГРАММИРОВАНИЯ
программы подготовки специалистов среднего звена
по специальности СПО
09.02.03 Программирование в компьютерных системах
базовой подготовки

Магнитогорск, 2017

ОДОБРЕНО
Предметно-цикловой комиссией «*Информатики и вычислительной техники*»

Председатель *И.Г. Зорина*

Протокол № 7 от 14 марта 2017 г.

Методической комиссией МпК
Протокол №4 от «23» марта 2017г

Разработчик:
преподаватель ФГБОУ ВО «МГТУ им. Г.И. Носова» Многопрофильный колледж Л. А. Фетисова

Комплект контрольно-оценочных средств для текущего контроля и промежуточной аттестации по учебной дисциплине составлен на основе ФГОС СПО по специальности 09.02.03 Программирование в компьютерных системах, утвержденного «28» июля 2014 г. № 804, и рабочей программы учебной дисциплины "Основы программирования"

ОБЩИЕ ПОЛОЖЕНИЯ

Учебная дисциплина «ОСНОВЫ ПРОГРАММИРОВАНИЯ» относится к общепрофессиональным дисциплинам профессионального цикла.

В результате освоения дисциплины обучающийся должен уметь:
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;

В результате освоения дисциплины обучающийся должен знать:
- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования.

Содержание дисциплины ориентировано на подготовку студентов к освоению профессиональных модулей ОПОП по специальности и овладению профессиональными компетенциями:
- ПК 1.1. Выполнять разработку спецификаций отдельных компонент.
- ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
- ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.
- ПК 1.4. Выполнять тестирование программных модулей.
- ПК 1.5. Осуществлять оптимизацию программного кода модуля.
- ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.

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 в команде, эффективно общаться с коллегами, руководством, потребителями.
- ОК 7. Брать на себя ответственность за работу членов команды (подчиненных), за 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.

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

Промежуточная аттестация в форме экзамена.

Оценка индивидуальных образовательных достижений по результатам текущего контроля успеваемости 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

Таблица 1

Паспорт оценочных средств

№	Контролируемые разделы (темы) учебной дисциплины	Контролируемые умения, знания	Контролируемые компетенции	Наименование оценочного средства	
				Текущий контроль	Промежуточная аттестация
	Раздел 2. Типы данных				экзамен
1	Тема 2.3. Ввод и вывод данных	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Контрольная работа	
	Раздел 3. Базовые конструкции изучаемых языков программирования				
2	Тема3.1. Операторы ветвления	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Контрольная работа	
3	Тема3.2. Операторы цикла	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Контрольная работа	
4	Тема3.4. Массивы	У1-2, З1-3	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Контрольная работа	
5	Тема 3.5. Строки	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Рубежный контроль	
	Раздел 4 Принципы структурного и модульного программирования				
6	Тема4.1. Конструкции структурного программирования	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Контрольная работа	
7	Тема4.4. Области действия идентификаторов	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Рубежный контроль	
	Раздел 5 Принципы объектно-ориентированного программирования				
8	Тема 5.2. Конструкторы	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Контрольная работа	
9	Тема 5.1. Классы.	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Тестовый контроль	

10	Тема5.3. Деструкторы	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.		
11	Тема5.5. Шаблоны классов	У1-2, З1-5	ПК 1.1., ПК 1.2. , ПК 1.3., ПК 1.4. ПК 1.5. ПК 3.1.	Тестовый контроль	

1. ВХОДНОЙ КОНТРОЛЬ

Спецификация

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

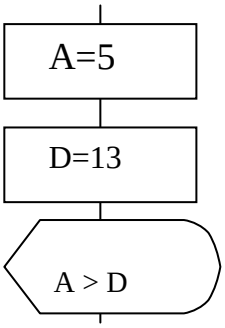
- Математика
- Информатика

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

ТЕСТ- В1

1. Определить, какого типа будет выражение **- 15/3**
   - **B. Логического**
   - **C. Символьного**
   - **D. Вещественного**
   - **E. Целого**
2. Какое значение переменной S получится при вычислении **S= 20 mod 6**
   - **B. 3.3**
   - **C. 3**
   - **D. 2**
   - **E. 3.33**
3. Любая блок - схема начинается с блока**:**
   - **B. Ввода**
   - **C. Начало**
   - **D. Вывода**
   - **E. Вычисления**
4. Определить, какой оператор является оператором цикла:
   - **B. While do**
   - **C. If then**
   - **D. Case of**
   - **E. Then else**
5. Определить, что будет напечатано на экране в результате выполнения программы:
   - **B. 0**
   - **C. 13**
   - **D. 5**
   - **E. 1**



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. Данный вид контроля должен стимулировать 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 ТЕСТОВЫЙ КОНТРОЛЬ

Тема 5.1. Классы.
Для чего нужны классы?
Варианты ответа*:
*1) для определения новых типов в программе; 2) для упрощения работы с функциями; *3) для соединения данных и операций над ними; 4) для упрощения работы с указателями.

Тема 5.5. Шаблоны классов
1. Отметьте все утверждения, которые считаете верными:
1) нельзя с помощью шаблона создать функцию с таким же именем, как у явно определенной функции;

2.2 КОНТРОЛЬНЫЕ РАБОТЫ

Раздел 2 Типы данных

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) or B 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.

Раздел 3. Базовые конструкции изучаемых языков программирования

Тема 3.1. Операторы ветвления

1. Услуги телефонной сети оплачиваются по следующему правилу: разговоры до *A* минут в месяц — *B* руб., а разговоры сверх установленной нормы оплачиваются из расчета Сруб, 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.
3. К финалу конкурса лучшего по профессии «Специалист электронного офиса» были допущены трое: Иванов, Петров, Сидоров. Соревнования проходили в три тура. Иванов в первом туре набрал m_1 баллов, во втором — n_1, в третьем — p_1,. Петров — m_2,, n_2, p_2 соответственно; Сидоров — m_3,, n_3,, p_3. Составить программу, определяющую, сколько баллов набрал победитель.

Тема 3.2. 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 $i=1,2,...,n$.

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

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

$$a_n = \frac{a_{n-1} + a_{n-2}}{2}, a_1 = 1, a_2 = 2$$

Тема 3.4. 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 $a_1, a_2, ..., a_n$. Среди них есть поожительные и отрицательные.
   Заменить нулями те числа, величина которых по модулю больше максимального числа: $(|a_i| > \max\{a_1, a_2, ..., a_n\})$
   Составить алгоритм и написать программу.

3. Дан массив действительных чисел. Среди них есть равные. Найти его первый максимальный элемент и заменить его нулем.

Составить алгоритм и написать программу.

Раздел 4 Принципы структурного и модульного программирования

Тема 4.1. Конструкции структурного программирования.

1. Дан массив D, состоящий из N элементов.

Составить алгоритм и написать программу для вычисления суммы трех последовательно расположенных элементов массива с номерами от K до M.
В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.

2. Дано: три числа- a,b,c.

Составить алгоритм и написать программу для нахождения суммы большего и меньшего из трех чисел. В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.
В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.

3.Дано: формула $X(N) = \frac{1}{N!}$ Составить алгоритм и написать программу для формирования массива *X(N),* N-й член которого определяется формулой $X(N) = \frac{1}{N!}$
В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.

Раздел 5 Принципы объектно-ориентированного программирования

Тема 5.2. Конструкторы

1. Составить описание класса для представления комплексных чисел с возможностью задания вещественной и мнимой чаете;: как числами типов double, так и целыми числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.
2. Составить описание класса для работы с цепными спискам; строк (строки произвольной длины) с операциями включения в список, удаления из списка элемента с заданным значением данного, удаления конца списка, начиная с заданного элемента.
3. Составить описание класса для работы с цепными спискам; строк (строки произвольной длины) с операциями включения в список, удаления из списка элемента с заданным значением данного, удаления всего списка начиная с заданного элемента.

2.3 РУБЕЖНЫЙ КОНТРОЛЬ

Рубежный контроль охватывает содержание разделов Этапы решения задачи на компьютере, Типы данных, Базовые конструкции изучаемых языков программирования, Принципы структурного и модульного программирования. Количество контрольных точек определяется рабочей программой учебной дисциплины.

На каждой рубежной контрольной работе проверяются:

- *теоретические знания*, полученные обучающимися на лекциях, практических занятиях, при чтении учебников и учебных пособий во время внеаудиторной самостоятельной работы;
- *практические навыки*, полученные обучающимися на практических занятиях.

Примеры заданий рубежного контроля
Раздел1-Раздел 4

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

Критерии оценки

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

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.

Контрольные вопросы и задания экзамена

№	Контрольные вопросы	Тема
1	Отличительные характеристики операторов условного и безусловного переходов	Тема 2.3. Ввод и вывод данных Тема 3.1. Операторы ветвления Тема 3.2. Операторы цикла Тема 3.4. Массивы Тема 3.5. Строки Тема 4.1. Конструкции структурного программирования
2	Оператор безусловного перехода.	
3	Отличительные характеристики операторов условного и безусловного переходов	
4	Оператор цикла с последующим условием.	
5	Типы данных. Целый тип.	
6	Условный оператор. Простая ветка.	
7	Структура построения программы на языке С++	
8	Отличительные особенности оператора цикла с предусловием от цикла с постусловием.	
9	Массивы.	
10	Массивы. Двумерный массив	
11	Цикл в цикле.	
12	Массивы. Одномерный массив	
13	Цикл с параметром.	
14	Оператор цикла с предварительным условием.	
15	Условный оператор	
16	Оператор выбора	
17	Оператор цикла с последующим условием.	
18	Оператор цикла с предварительным условием	
19	Условный оператор. Сложная ветка.	
20	Отличительные особенности оператора цикла с предусловием от цикла с постусловием.	
21	Типы данных.	

№	Типовые задания	Тема
1	Дан двумерный массив, состоящий из четного числа строк. Поменять местами его строки следующим образом: первую строку поменять местами с последней, вторую с предпоследней и т.д. Работу с матрицей оформить в виде подпрограммы	Тема 3.4. Массивы

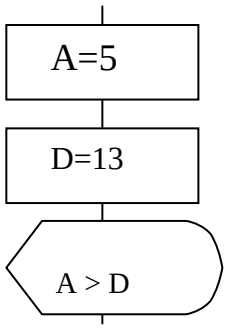
Критерии оценки

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

ПРИЛОЖЕНИЕ 1

ТЕСТОВЫЕ ЗАДАНИЯ ВХОДНОГО КОНТРОЛЯ
ТЕСТ- В1

6. Определить, какого типа будет выражение **- 15/3**
   - **B. Логического**
   - **C. Символьного**
   - **D. Вещественного**
   - **E. Целого**
7. Какое значение переменной S получится при вычислении **S= 20 mod 6**
   - **B. 3.3**
   - **C. 3**
   - **D. 2**
   - **E. 3.33**
8. Любая блок - схема начинается с блока**:**
   - **B. Ввода**
   - **C. Начало**
   - **D. Вывода**
   - **E. Вычисления**
9. Определить, какой оператор является оператором цикла:
   - **B. While do**
   - **C. If then**
   - **D. Case of**
   - **E. Then else**
10. Определить, что будет напечатано на экране в результате выполнения программы:
    - **B. 0**
    - **C. 13**
    - **D. 5**
    - **E. 1**



ТЕСТ- В2

1. Определить, какого типа будет выражение **- 1+0.0**
   - **A. Логического**
   - **B. Символьного**
   - **C. Вещественного**
   - **D. Целого**
2. Какое значение переменной S получится при вычислении **S= 20 div 6**
   - **A. 3.3**
   - **B. 3**
   - **C. 2**
   - **D. 3.33**
3. Любая блок - схема начинается с блока**:**
   - **A. Ввода**
   - **B. Начало**
   - **C. Условия**

D. Вычисления

4. Определить, какое значение переменной Y получится при вычислении логического выражения – **Y=odd(trunc(10*0.182))**
   - **A. 0.182**
   - **B. 0**
   - **C. 1.82**
   - **D. 1**
5. Определить, как будет выглядеть запись арифметического выражения при использовании стандартных функций – **Y=** x^4
   - **A. Y=sqrt(sqr(x))**
   - **B. Y=sqr(ln(x))**
   - **C. Y=exp(ln(x))**
   - **D. Y=exp(4*ln(x))**

ТЕСТ- В3

1. Определить, какого типа будет выражение - **SQR(16)**
   - **A. Вещественный**
   - **B. Символьный**
   - **C. Логический**
   - **D. Целый**
2. Какое значение переменной S получится при вычислении **S= 3 div 7**
   - **A. 3**
   - **B. 0.4**
   - **C. 0**
   - **D. 0.42**

3. Программа заканчивается служебным словом:
   - **A. AND**
   - **B. VAR**
   - **C. OR**
   - **D. END**

4. Определить, какое значение переменной Y получится при вычислении логического выражения – **Y= a or not(b) при a=false, b=true**
   - **A. 0**
   - **B. 1**
   - **C. 2**
   - **D. -1**

5. Определить, как будет выглядеть запись арифметического выражения при использовании стандартных функций – **Y=e^{x+1}**
   - **A. Y=exp(ln(x+1))**
   - **B. Y=exp(x+1)**
   - **C. Y=sqr(ln(x+1))**
   - **D. Y=exp((x+1)*ln(x))**

ТЕСТ- В4

1. Определить, какого типа будет выражение - **Sin(0)**
   - **A. Логический**
   - **B. Целый**
   - **C. Вещественный**
   - **D. Символьный**
2. Какое значение переменной S получится при вычислении **S= 3 mod 7**
   - **A. 3**
   - **B. 0.4**

C. 0
D. 0.42

3. Переход на начало программы осуществляется с помощью оператора:

A. CASE
B. REPEAT
C. GOTO
D. UNTIL

4. Определить, какое значение переменной Y получится при вычислении логического выражения – **Y= not odd (b) при b=25**

A. 0
B. 25
C. 1
D. 24

5. Определить, как будет выглядеть запись арифметического выражения при использовании стандартных функций – **Y=ln(x+1)**

A. Y=exp(ln(x+1))
B. Y=ln(x+1)
C. Y=sqr(ln(x+1))
D. Y=exp((x+1)*ln(x))

ТЕСТ- В5

1.Определить, какой оператор является оператором цикла:

A. For do
B. If then
C. Case of
D. Begin end

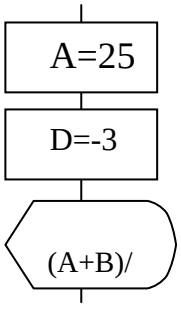
2. Определить, что будет напечатано в результате выполнения фрагмента рограммы:

A. 8
B. 10
C. 5
D. 3

```
c:=1; b:=20;
d:=sqrt(b-4*c);
writeln( (b/2);
```

3. Определить, что будет напечатано на экране в результате выполнения программы:

A. 11
B. 13
C. 14
D. 10



4. Определить, какое значение переменной S получится в результате выполнения фрагмента программы:

A. 9
B. 6
C. 5
D. 3

```
a:=3; b:=3;
if a<b then s:=a*a
else s:=a*b;
```

5**.** Определить, как будет выглядеть запись арифметического выражения при использовании стандартных функций – $\mathbf{Y=e^{x+1}}$

A. **Y=exp(ln(x+1))**
B. **Y=exp(x+1)**
C. **Y=sqr(ln(x+1))**
D. **Y=exp((x+1)*ln(x))**

ТЕСТ- В6

1. Какое значение переменной S получится при вычислении **S= 20 div 6**
   E. **3.3**
   F. **3**
   G. **2**
   H. **3.33**
2. Любая блок - схема начинается с блока**:**
   E. **Ввода**
   F. **Начало**
   G. **Условия**
   H. **Вычисления**
3. Определить, какое значение переменной Y получится при вычислении логического выражения – **Y=odd(trunc(10*0.182))**
   E. **0.182**
   F. **0**
   G. **1.82**
   H. **1**
4. Определить, как будет выглядеть запись арифметического выражения при использовании стандартных функций – **Y=** x^4
   E. **Y=sqrt(sqr(x))**
   F. **Y=sqr(ln(x))**
   G. **Y=exp(ln(x))**
   H. **Y=exp(4*ln(x))**
5. Определить, какого типа будет выражение **- 1+0.0**
   E. **Логического**
   F. **Символьного**
   G. **Вещественного**
   H. **Целого**

ПРИЛОЖЕНИЕ 2

Классы.
Для чего нужны классы?
Варианты ответа*:
*1) для определения новых типов в программе; 2) для упрощения работы с функциями; *3) для соединения данных и операций над ними; 4) для упрощения работы с указателями.
2. Методы класса определяют:
*1) какие операции можно выполнять с объектами данного класса; 2) какие типы значений могут принимать данные-члены класса; 3) каким образом реализуется защита данных-членов класса.
3. Атрибуты (данные-члены) класса могут быть:
1) только целыми числами; 2) любыми встроенными типами; *3) любого определенного в программе типа.
4. Какая из записей соответствует обращению к атрибуту arg класса A в определении метода этого же класса?
Варианты ответа:
*1) this –> arg; *2) arg; 3) A –> arg; 4) A –> this –> arg.
5. Если имеется код
class A {public: int a; }; A *obj;
то как обратиться к переменной a?
Варианты ответа:
1) obj.a; 2) (*obj) –> a; *3) obj –> a; 5) obj :: a.
6. Если имеется код
class A {public: int a, b, c;}; A *obj;
то как обратиться к переменной b?
Варианты ответа:
1) (*obj) –> b; 2) A :: b; *3) (*obj).b; 4) obj –> a.b.
* Звездочкой в тестах отмечены правильные ответы.

Шаблоны классов
1. Отметьте все утверждения, которые считаете верными:
1) нельзя с помощью шаблона создать функцию с таким же именем, как у явно определенной функции;
2) цель введения шаблонов – создание функций, которые могут обрабатывать однотипные данные;
*3) при использовании шаблонов функций возможна перегрузка как с помощью шаблонов, так и функций;
*4) в качестве описания шаблона функции используется прототип шаблона:
template список _ параметров _шаблона
2. Можно ли задать шаблон со значением параметра по умолчанию?
Варианты ответа:
*1) да; 2) нет.
3. Каков правильный заголовок шаблона?
Варианты ответа:
*1) template<class t1,class t2>; 2) template <class t1,t2>; 3) template <class t,class t>;
4. Сколько параметров может быть у шаблона при определении шаблона функции?
Варианты ответа:
1) 1; 2) столько, сколько аргументов у функции; *3) столько, сколько типов используется для параметризации.
5. Отметьте правильный вариант описания шаблона семейства функций:

1) template(class T)
void func(T* p1,T* p2){…}
2) template <class T>;
void func(T* p1,T* p2){…}
*3) template<class T>
void func(T* p1,T* p2){…}
4) template<class T>
void func(T* p1,T* p2){…}
6. Можно ли использовать класс-шаблон в качестве базового класса?
Варианты ответа:
*1) да; 2) нет.

ПРИЛОЖЕНИЕ 3

По специальности 09.02.03 Программирование в компьютерных системах базового уровня подготовки по программе учебной дисциплиныОП.05 ОСНОВЫ ПРОГРАММИРОВАНИЯ предусмотрены контрольные работы.

Раздел 2 Типы данных

КОНТРОЛЬНАЯ РАБОТА № 1
Вариант1

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) or B 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.

Вариант2

1. Какие из приведенных ниже записей вещественных чисел являются неправильными и почему?:
   а. .0E-2 в. -5.3E4
   б. 0.1E-5 г. +2.3E+3

2. Вычислить значения выражений, если A=25, B=7.8, C=-17,3, M=5, x=8,7 E=true
   а. (A+B)/C*M
   б. (A<B) и (x+A<B) или(C<M) или E

3. Организовать ввод и вывод данных заданных типов, снабдив распечатки соответствующими заголовками

T=’ ‘ L=35 A=’S’ Y=21.345

Дать протокол программы.

Вариант3

1. Можно ли вводить с помощью оператора READ значения булевого типа?

2. Определить тип результата следующих выражений
   а. sin(x)+2*cos(Y)+z
   б. I+sqrt(J),

если известно, что I,J-int; x,y,z, -float?

3. Организовать ввод и вывод данных заданных типов, снабдив распечатки соответствующими заголовками

R=7.5 B=-67.3 E='T' V=-4192.21

Дать протокол программы.

Раздел 3. Базовые конструкции изучаемых языков программирования

КОНТРОЛЬНАЯ РАБОТА № 2

Вариант 1

Услуги телефонной сети оплачиваются по следующему правилу: разговоры до A минут в месяц — B руб., а разговоры сверх установленной нормы оплачиваются из расчета Сруб, за минуту. Написать программу, вычисляющую плату за пользование телефоном для введенного времени разговоров за месяц.

Вариант 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.

Вариант 3

К финалу конкурса лучшего по профессии «Специалист электронного офиса» были допущены трое: Иванов, Петров, Сидоров. Соревнования проходили в три тура. Иванов в первом туре набрал m_1 баллов, во втором — n_1, в третьем — p_1,. Петров — m_2,, n_2, p_2 соответственно; Сидоров — m_3,, n_3,, p_3. Составить программу, определяющую, сколько баллов набрал победитель

Вариант 1

Составить алгоритм и написать программу.

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

Вариант 2

Составить алгоритм и написать программу.

Определить, сколько процентов от всего количества элементов последовательности целых чисел составляют нечетные элементы.

Вариант 3

С оставить алгоритм и написать программу.

Последовательность $a_1, a_2, \ldots, a_n$ состоит из нулей и единиц. Поставить в начало этой последовательности нули, а затем единицы.

КОНТРОЛЬНАЯ РАБОТА № 3

Вариант 1

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

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

Вариант 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 $i=1,2,...,n$.

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

Вариант 3

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

$$a_n = \frac{a_{n-1} + a_{n-2}}{2}, a_1 = 1, a_2 = 2$$

КОНТРОЛЬНАЯ РАБОТА № 4

Вариант 1

Составить алгоритм и написать программу.
В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах.

Вариант 2

Составить алгоритм и написать программу.

Даны действительные числа $a_1, a_2, ..., a_n$. Среди них есть положительные и отрицательные. Заменить нулями те числа, величина которых по модулю больше максимального числа: $(|a_i| > \max\{a_1, a_2, ..., a_n\})$

Вариант 3

Составить алгоритм и написать программу.
Дан массив действительных чисел. Среди них есть равные. Найти его первый максимальный элемент и заменить его нулем.

Раздел 4 Принципы структурного и модульного программирования

КОНТРОЛЬНАЯ РАБОТА № 5

Вариант 1

Составить алгоритм и написать программу для вычисления суммы факториалов всех четных чисел от *m* до *n.*

В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.

Вариант 2

Дано: формула $X(N) = \frac{1}{N!}$

Составить алгоритм и написать программу для формирования массива *X(N),* N-й член которого определяется формулой $X(N) = \frac{1}{N!}$

В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.

Вариант 3

Дано: натуральные числа *K* и *N.*
Составить алгоритм и написать программу для формирования массива *A,* элементами которого являются числа, сумма цифр которых равна *K* и которые не больше *N.*

В основной программе обеспечить ввод всех необходимых данных, обращение к подпрограмме и вывод результата на экран.

Раздел 5 Принципы объектно-ориентированного программирования

КОНТРОЛЬНАЯ РАБОТА № 6

Вариант1

Составить описание объектного типа TMatr, обеспечивающего размещение матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы.

Вариант2

Составить описание класса для представления комплексных чисел с возможностью задания вещественной и мнимой чаете;: как числами типов double, так и целыми числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

Вариант3

Составить описание класса для работы с цепными спискам; строк (строки произвольной длины) с операциями включения в список, удаления из списка элемента с заданным значением данного, удаления конца списка, начиная с заданного элемента.

ПРИЛОЖЕНИЕ 4

РУБЕЖНЫЙ КОНТРОЛЬ

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 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 5
Выполнение каждой программы на C++ начинается с функции_____.
1: void
2: include
3: main
4: using

Вопрос 6
Каждый оператор заканчивается_____.
1: endl
2: return
3: точкой с запятой
4: управляющей последовательностью
Вопрос 18
Элементы массива связаны тем, что они имеют одни и те же _____
1: имя
2: тип

3: значения
4: размерность

Вопрос 7
Число, используемое для обращения к отдельному элементу массива называется_____
1: тип
2: индекс
3: значение
4: позиция

Вопрос 8
Оператор-переключатель - это:
1: оператор для выбора одного из нескольких вариантов (SWITCH)
2: строка с меткой DEFAULT
3: CASE
4: BREAK

Вопрос 9
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 10
Формальный аргумент - это:
1: конкретное значение, присвоенное этой переменной вызывающей программой
2: переменная и вызываемой программе
3: строка, которая пишется в скобках функции