

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Кабардино-Балкарский государственный
университет им. Х.М. Бербекова» (КБГУ)

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

СОГЛАСОВАНО

Руководитель образовательной
программы  М.М. Лафинева

« 12 » 04 2023г.



« _____ 2023г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ (ОЦЕНОЧНЫХ МАТЕРИАЛОВ)
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)
«ЯЗЫКИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»

02.03.02 Фундаментальная информатика и информационные технологии
(код и наименование направления подготовки)

«Проектирование систем искусственного интеллекта»
(наименование профиля подготовки)

Бакалавр

Квалификация (степень) выпускника

Очная

Форма обучения

СОДЕРЖАНИЕ

1. Перечень компетенций и этапы их формирования	3
2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	4
3. Критерии формирования оценок на различных этапах их формирования	5
4. Методические материалы, определяющие процедуры оценивания результатов освоения образовательной программы	6

1. Перечень компетенций и этапы их формирования

Шифр и название компетенций:
профессиональных (ПКС):

Коды	Содержание компетенций
ПКС-3	Способен преподавать математику и информатику в средней школе, специальных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения

Общая характеристика компетенции

Тип компетенции: общепрофессиональная компетенция выпускника образовательной программы по направлению подготовки высшего образования 02.03.02 Фундаментальная информатика и информационные технологии, уровень ВО бакалавр.

1.1. Этапы формирования компетенций и средства оценивания

Результаты обучения (компетенции)	Индикаторы достижения компетенции	Освоенные показатели оценки результатов обучения	Виды оценочного материала, обеспечивающий формирование компетенций
ПКС-3. Способен преподавать математику и информатику в средней школе, специальных учебных заведениях на основе полученного фундаментального образования и научного мировоззрения	ПКС-3.1. Способен использовать методические основы преподавания профессиональных дисциплин	ПКС-3.1. 3-1. методические основы преподавания дисциплин математики и информатики. ПКС-3.1. У-1. Профессионально грамотно пользоваться организационно-методическим и учебно-методическим обеспечением образовательной программы соответствующего уровня. ПКС-3.1. В-1. психолого-педагогическими и методическими основами преподавания дисциплин математики и информатики.	Типовые оценочные материалы для устного опроса (п. 5.1.1); типовые оценочные материалы для контрольной работы (п. 5.2.1); типовые тестовые задания (п. 5.2.2); типовые оценочные материалы к экзамену (п. 5.2.3)
	ПКС-3.2. Способен планировать лекционные и семинарские занятия по программам профессионального обучения математике и информатике, с учетом	ПКС 3.2. 3-1. образовательный стандарт и программы среднего общего образования, среднего профессионального образования и дополнительные	

	уровня подготовки и психологию аудитории	<p>общеобразовательные и профессиональные программы соответствующего уровня</p> <p>ПКС 3.2. У-1. Применять теоретические и практические основы по дополнительным разделам элементарной математики при проведении методических и экспертных работ.</p> <p>ПКС-3.2. В-1. Способностью к планированию и осуществлению педагогической деятельности с учетом специфики предметной области в образовательных организациях.</p>	
--	--	--	--

2. Показатели и критерии оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Первый этап (уровень) <i>36-50 баллов</i>	Второй этап (уровень) <i>51-60 баллов</i>	Третий этап (уровень) <i>61-70 баллов</i>
<p>На данном уровне обучающийся запоминает и воспроизводит изученный материал. Студент: знает (запоминает и воспроизводит) употребляемые термины; знает конкретные факты; знает методы и процедуры; знает основные понятия; знает правила и принципы.</p>	<p>На данном этапе обучающийся понимает значение изученного материала, может преобразовать материал из одной формы выражения в другую. В качестве показателя понимания может также выступать интерпретация материала студентом (объяснение, краткое изложение) или же предположение о дальнейшем ходе явлений, событий (предсказание последствий, результатов). Обучающийся: понимает факты, правила и принципы; интерпретирует словесный материал, схемы, графики, диаграммы; преобразует словесный материал в математические выражения; предположительно описывает будущие последствия, вытекающие из имеющихся данных.</p>	<p>Этот уровень обозначает умение использовать изученный материал в конкретных условиях и в новых ситуациях. Сюда входят применение правил, методов, понятий, законов, принципов, теорий. Соответствующие результаты обучения требуют более высокого уровня владения материалом, чем понимание. Студент: использует понятия и принципы в новых ситуациях; применяет законы, теории в конкретных практических ситуациях; демонстрирует правильное применение метода или процедуры.</p>

3. Критерии формирования оценок на различных этапах их формирования

Распределение баллов текущего и рубежного контроля

Вид работы	Трудоёмкость часов / зачётных единиц		
	1 семестр	2 семестр	всего
Общая трудоёмкость (в часах)	180	252	432
Контактная работа (в часах):	51	90	141
<i>Лекционные занятия (Л)</i>			
<i>Практические занятия (ПЗ)</i>	17	36	53
<i>Семинарские занятия (СЗ)</i>			
<i>Лабораторные работы (ЛР)</i>	34	54	88
Самостоятельная работа (в часах), в том числе контактная (внеаудиторная)	102	135	237
Расчетно-графическое задание			
Реферат (Р)			
Эссе (Э)			
Контрольная работа (К)			
Самостоятельное изучение разделов	102	135	237
Курсовой проект (КП), курсовая работа (КР)			
Подготовка и прохождение промежуточной аттестации	27	27	54
Вид промежуточной аттестации	Экзамен	Экзамен	Экзамен

Критерии формирования оценок по промежуточной аттестации (для экзамена в случае, если экзаменационный билет содержит два вопроса)

Семестр	Шкала оценивания (по итогам текущего и рубежного контроля)			
	Неудовлетворит. (36-60 баллов)	Удовлетворит. (61-80 баллов)	Хорошо (81-90 баллов)	Отлично (91-100 баллов)
1, 2	Обучающийся имеет 36-60 баллов по итогам текущего и рубежного контроля, на экзамене не дал полного ответа ни на один вопрос. Обучающийся имеет 36-45 баллов по итогам текущего и рубежного контроля, на экзамене дал полный ответ только на один вопрос	Обучающийся имеет 36-50 баллов по итогам текущего и рубежного контроля, на экзамене дал полный ответ на один вопрос и частично (полностью) ответил на второй. Обучающийся имеет 46-60 баллов по итогам текущего и	Обучающийся имеет 51-60 баллов по итогам текущего и рубежного контроля, на экзамене дал полный ответ на один вопрос и частично (полностью) ответил на второй. Обучающийся имеет 61 – 65 баллов по итогам текущего и рубежного	Обучающийся имеет 61-70 баллов по итогам текущего и рубежного контроля, на экзамене дал полный ответ на один вопрос и частично (полностью) ответил на второй.

		рубежного контроля, на экзамене дал полный ответ на один вопрос или частично ответил на оба вопроса. Обучающийся имеет по итогам текущего и рубежного контроля 61-70 баллов на экзамене не дал полного ответа ни на один вопрос	контроля, на экзамене дал полный ответ на один вопрос и частично ответил на второй. Обучающийся имеет 66-70 баллов по итогам текущего и рубежного контроля, на экзамене) дал полный ответ только на один вопрос.	
--	--	--	---	--

4. Методические материалы, определяющие процедуры оценивания результатов освоения образовательной программы

Перечень оценочных средств

№	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
1.	Коллоквиум	Средство контроля усвоения учебного материала темы, раздела или разделов дисциплины, организованное как учебное занятие в виде собеседования преподавателя с обучающимися.	Вопросы по темам/разделам дисциплины
2.	Тест	Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося.	Фонд тестовых заданий

Перечень вопросов для проведения коллоквиума (контролируемая компетенция ПКС-3)

Тема1. Введение. Основные понятия языков программирования. Среды программирования

1. Организация входа в интегрированную среду
2. Этапы разработки программы
3. Основное меню среды программирования
4. Открытие и сохранение файла в среде программирования
5. Организация поиска нужного фрагмента программы
6. Компиляция и запуск программы в среде программирования
7. Организация работы с точками останова
8. Установка необходимых параметров интегрированной среды
9. Получение справочной информации
10. Полный цикл работ по набору, отладке и исполнению примеров программ с применением средств пошаговой отладки среды программирования.

Тема 2. Базисные типы данных и операторный базис в процедурных языках программирования. Методы программирования с использованием статических структур данных

1. Вычислите длину окружности, площадь круга и объём шара одного и того же заданного радиуса.
2. Определите номер квадранта, в котором находится точка с заданными координатами (x, y) .
3. Подсчитайте число и сумму положительных, число и произведение отрицательных элементов заданного массива $A(N)$.
4. Запишите подряд в массив $A(N)$ элементы заданного массива $B(2N)$, стоящие на чётных местах, а элементы, стоящие на нечетных местах, запишите в массив $C(N)$.
5. Дана матрица $A(N, M)$. Найдите её наибольший элемент и номера строки и столбца, на пересечении которых он находится.
6. Дана матрица $A(N, N)$ и целое P . Преобразуйте матрицу по правилу: строку с номером P сделайте столбцом с номером P , а столбец с номером P сделайте строкой с номером P .
7. Выясните, имеются ли в заданном векторе $A(N)$ два подряд идущих нулевых элемента.
8. Все элементы заданного вектора $A(N)$, начиная с первого по порядку положительного элемента, уменьшите на единицу.
9. В заданной целочисленной матрице $A(N, M)$ выведите на печать индексы первого положительного элемента, кратного заданному числу K . Если таких элементов в матрице нет, то выведите соответствующий текст. Элементы матриц просматривайте слева направо и сверху вниз.
10. Элементы заданной матрицы $A(N, N)$ переписывайте построчно в одномерный массив до тех пор, пока не встретится нулевой элемент.
11. В заданной матрице $A(N, M)$ найдите количество строк, не содержащих отрицательных чисел.
12. Дана матрица $A(N, M)$. Постройте вектор $B(N)$, элементы B_i которого равны единице, если элементы i -й строки образуют упорядоченную по убыванию или по возрастанию последовательность, и нулю во всех остальных случаях.

Тема 3. Процедуры и функции

1. Из заданного множества точек на плоскости выберите две различные точки так, чтобы количества точек, лежащих по разные стороны прямой, проходящей через эти две точки, различались наименьшим образом.

2. Определите радиус и центр окружности, на которой лежит наибольшее число точек заданного на плоскости множества точек.

3. Задано множество M точек на плоскости. Определите, верно ли, что для каждой точки $A \in M$ существует точка $B \in M$ ($A \neq B$) такая, что не существует двух точек множества M , лежащих по разные стороны от прямой AB .

4. Определите радиус и центр такой окружности, проходящей хотя бы через три различные точки заданного множества точек на плоскости, что минимальна разность количеств точек, лежащих внутри и вне окружности.

5. В множестве точек на плоскости найдите пару точек с максимальным расстоянием между ними.

6. Расстояние между двумя множествами точек – это расстояние между наиболее близко расположенными точками этих множеств. Найдите расстояние между двумя заданными множествами точек на плоскости.

7. Многоугольник (не обязательно выпуклый) задан на плоскости перечислением координат вершин в порядке обхода его границы. Определите площадь многоугольника.

8. Задано множество точек M в трехмерном пространстве. Найдите такую из них, что шар заданного радиуса с центром в этой точке содержит максимальное число точек из M .

9. Задано множество прямых на плоскости (коэффициентами своих уравнений). Подсчитайте количество точек пересечения этих прямых.

10. В трехмерном пространстве задано множество материальных точек. Найдите ту из них, которая расположена наиболее близко к центру тяжести этого множества.

Тема 4. Рекурсивные методы в программировании

1. Посредством рекурсивной функции $\text{summ}(s, i, j)$ проверьте, является ли симметричной часть строки s , начинающаяся i -м и заканчивающаяся j -м её элементами.
2. «Переверните» заданное натуральное число, используя рекурсивную функцию.
3. Найдите минимальный элемент массива $A(N)$, используя рекурсию.
4. Найдите наибольший общий делитель двух натуральных чисел. Используйте рекурсию.
5. Проверьте, является ли заданное натуральное число простым. Используйте рекурсивную функцию или процедуру.
6. Реализуйте рекурсивный алгоритм построения цепочки из имеющегося набора костей домино.
7. Задан набор слов. Рекурсивно постройте из них цепочку максимальной длины таким образом, чтобы символ в конце слова совпадал с символом в начале следующего.
8. Выполните перевод числа из десятичной системы счисления в двоичную, используя рекурсивную процедуру/функцию.
9. Посредством рекурсивной функции определите количество единиц в двоичном представлении натурального числа.
10. Подсчитайте количество различных представлений заданного натурального m в виде суммы не менее двух попарно различных положительных слагаемых. Представления, различающиеся лишь порядком слагаемых, считаются одинаковыми.
11. Напишите рекурсивную программу, которая в заданном дереве по заданному n считает число всех вершин высоты n и печатает список этих вершин (в том или ином порядке).
12. Напишите рекурсивную программу, которая печатает все перестановки чисел $1, \dots, n$ по одному разу.
13. Перечислите все представления положительного целого числа n в виде суммы последовательности невозрастающих целых положительных слагаемых. Используйте рекурсивную процедуру.
14. Напишите рекурсивную программу обхода дерева во внутреннем порядке.

Тема 5. Понятие файла. Методы упорядочения файлов

1. Создайте файл из 50 целых чисел, используя генератор случайных чисел. Выведите эти числа на экран. Выведите на экран все числа из этого файла, оканчивающиеся на 7.

Найдите среднее арифметическое всех чисел, входящих в созданный файл. Найдите наименьшее из них.

2. Создайте два упорядоченных файла целых чисел. Объедините их, создав новый упорядоченный файл.

3. Создайте текстовый файл. Подсчитайте количество предложений в нем. Найдите самое длинное слово.

1. Создайте файл из 55 целых чисел, используя генератор случайных чисел. Перепишите все четные числа из этого файла в другой. Выведите оба файла на экран. Найдите в новом файле среднее арифметическое его элементов. Найдите наибольший элемент в новом файле.

2. Используя генератор случайных чисел, создайте два упорядоченных по возрастанию файла: первый – из четных чисел, второй – из нечетных чисел.

3. Создайте текстовый файл. Подсчитайте количество знаков препинания в нем. Найдите самое короткое слово.

1. Используя генератор случайных чисел, создайте файл из 20 целых чисел, кратных 5. Выведите его на экран. Найдите среднее арифметическое этих чисел. Найдите количество элементов, оканчивающихся нулем. Найдите наименьший из таких элементов.

2. Создайте два упорядоченных по убыванию файла. Объедините их в упорядоченный по убыванию файл.

3. Создайте текстовый файл. Уберите лишние пробелы. Выведите на экран слова, начинающиеся с буквы «А»

1. С помощью датчика случайных чисел создайте файл из 35 целых чисел. Выведите его на экран. Найдите наибольшее из чисел, оканчивающихся на 3. Подсчитайте количество чисел, оканчивающихся на 3. Найдите среднее арифметическое остальных чисел.

2. С помощью датчика случайных чисел создайте два упорядоченных по убыванию файла. Объедините их в упорядоченный по возрастанию файл.

3. Создайте текстовый файл. Определите количество вопросительных предложений и количество слов в первом предложении.

1. С помощью датчика случайных чисел создайте файл из 40 целых чисел. Выведите все четные числа, встречающиеся в созданном файле. Подсчитайте их количество. Найдите их среднее арифметическое. Найдите число, ближайшее к среднему арифметическому.

2. С помощью датчика случайных чисел создайте два упорядоченных файла: один – по убыванию, другой – по возрастанию.

3. Создайте текстовый файл. Зашифруйте его, используя метод обратного чтения.

1. Создайте файл из 20 вещественных чисел, выбранных случайным образом из числового отрезка $[10, 30]$. Выведите его на экран. Найдите наибольший и наименьший элементы в этом файле. Найдите сумму всех элементов файла.

2. С помощью датчика случайных чисел создайте два упорядоченных файла: один – по неубыванию, другой – по невозрастанию. Объедините эти файлы в новый, упорядоченный по убыванию файл.

3. Создайте текстовый файл. Выведите его на экран. Выведите на экран последнее предложение. Определите количество слов в нем.

1. Создайте файл из 30 вещественных чисел, выбранных случайным образом из числового отрезка $[5, 8]$. Выведите его на экран. Найдите среднее арифметическое его элементов. Найдите все элементы, расстояние которых от среднего арифметического не превышает 1, и запишите их в новый файл.

2. С помощью датчика случайных чисел создайте два упорядоченных по возрастанию файла. Объедините их в упорядоченный по убыванию файл.

3. Создайте текстовый файл. Выведите его на экран. Определите, есть ли в тексте слово «утро».

1. Создайте файл из 25 вещественных чисел, выбранных случайным образом из числового отрезка $[5, 7]$. Найдите наибольший и наименьший элементы. Найдите среднее арифметическое первых десяти элементов.

2. Создайте два файла. Первый – по закону $x_i = 1/i$, второй – $y_i = 1/(i+1)$, где $i = 1, \dots, 20$. Объедините их в упорядоченный по возрастанию файл.

3. Создайте текстовый файл. Выведите его на экран. Поменяйте местами первое и последнее слово.

1. Создайте файл из 50 вещественных чисел, выбранных случайным образом из числового отрезка $[1, 4]$. Выведите файл на экран. Найдите наибольший элемент. Найдите среднее арифметическое последних 15 чисел.

2. Создайте два файла. Первый – по закону $x_i = i^2$, где $i = 1, \dots, 10$, а второй – по закону $y_k = k^2 + 1$, где $k = 1, \dots, 15$. Создайте новый файл, упорядоченный по убыванию, путем слияния двух ранее созданных.

3. Создайте текстовый файл. Выведите его на экран. Подсчитайте, сколько в этом файле слов, оканчивающихся на букву «д».

1. Создайте файл из 25 вещественных чисел, выбранных случайным образом из числового отрезка $[1, 8]$. Выведите файл на экран. Найдите наименьший элемент. Найдите

среднее арифметическое всех элементов. Выведите на экран все элементы, которые отличаются от среднего арифметического не более чем на 0,5.

2. Создайте два файла. Первый – по правилу $x_i = 3i + 1$, где $i = 5, \dots, 15$, второй – по правилу $y_k = 2k - 1$, где $k = 4, \dots, 18$. Получите новый файл путем слияния двух созданных. Упорядочьте его элементы по возрастанию.

3. Создайте текстовый файл. Выведите его на экран. Выведите на экран самое длинное предложение файла.

Тема 6. Понятие модуля. Стандартные библиотечные модули CRT и GRAPH. Пользовательские модули

Разработайте тест модуля и использующую ее программу для решения следующих задач:

Задача 1

1. Создайте файл из 30 вещественных чисел, выбранных случайным образом из числового отрезка $[1, 5]$. Найдите его наибольший элемент. Найдите среднее арифметическое 18 последних чисел, среднее арифметическое всех чисел.

2. Создайте два файла. Первый – по правилу $x_i = 2i + 1$, где $i = 1, \dots, 9$, второй – по правилу $y_k = 1 - 2k$, где $k = 1, \dots, 20$. Получите новый файл путем слияния двух созданных. Упорядочьте его элементы по возрастанию.

3. Создайте текстовый файл. Выведите его на экран. Выведите на экран второе и предпоследнее предложения.

Задача 2

1. Используя датчик случайных чисел, создайте файл из 28 целых чисел, принадлежащих числовому отрезку $[30, 100]$. Выведите его на экран. Найдите среднее арифметическое всех нечетных чисел. Найдите наименьшее среди четных чисел.

2. Создайте два файла. Первый – по правилу $x_i = (1 - 2i)/i$, где $i = 1, \dots, 20$, второй – по правилу $y_k = 1/k$, где $k = 1, \dots, 10$. Получите новый файл путем слияния двух созданных. Упорядочьте его элементы по убыванию.

3. Создайте текстовый файл. Выведите его на экран. Выведите на экран все предложения, начинающиеся с буквы «Ц».

Задача 3

1. Дан файл f , компоненты которого являются действительными числами. Найдите: а) сумму компонент файла f ; б) произведение компонент файла f ; в) сумму квадратов компонент файла f ; г) модуль суммы и квадрат произведения компонент файла f ; д) последнюю компоненту файла.

2. Даны символьные файлы f и g . Определите, совпадают ли компоненты файла f с компонентами файла g . Если нет, то получите номер первой компоненты, в которой файлы f и g различаются между собой.

3. Дан текстовый файл f . Получите все его строки, содержащие более 60 символов.

Задача 4

1. Дан файл *f*, компоненты которого являются действительными числами. Найдите: а) наибольшее из значений компонент; б) наименьшее из значений компонент с четными номерами; в) наибольшее из значений модулей компонент с нечетными номерами; г) сумму наибольшего и наименьшего из значений компонент; д) разность первой и последней компонент файла.

2. Дан символьный файл *f*. Получите его перевернутую копию в файле *g*.

3. Дан текстовый файл *f*. Перепишите в файл *g* все компоненты файла *f* с заменой в них символа «0» на символ «1» и наоборот.

Задача 5

1. Дан файл *f*, компоненты которого являются целыми числами. Найдите: а) количество четных чисел среди компонент; б) количество удвоенных нечетных чисел среди компонент; в) количество квадратов нечетных чисел среди компонент.

2. Даны символьные файлы *f1* и *f2*. Перепишите с сохранением порядка следования компоненты файла *f1* в файл *f2*, а компоненты файла *f2* – в файл *f1*. Используйте вспомогательный файл *h*.

3. Дан текстовый файл *f*. Запишите в перевернутом виде строки файла *f* в файл *g*. Порядок строк в файле *g* должен: а) совпадать с порядком исходных строк в файле *f*; б) быть обратным по отношению к порядку строк исходного файла.

Задача 6

1. Дан файл *f*, компоненты которого являются целыми числами. Получите в файле *g* все компоненты файла *f*:

а) являющиеся четными числами;

б) делящиеся на 3 и не делящиеся на 7;

в) являющиеся точными квадратами.

2. Дан символьный файл *f*. Добавьте в его конец символы «e», «n», «d» (если это необходимо, используйте дополнительный файл *g*).

3. Дан текстовый файл *f*. Перепишите компоненты файла *f* в файл *g*, вставляя в начало каждой строки по одному пробелу. Порядок компонент должен быть сохранен.

Задача 8

1. Дан файл *f*, компоненты которого являются целыми числами. Запишите в файл *g* все четные числа файла *f*, а в файл *h* – все нечетные. Порядок следования чисел сохраняется.

2. Дан символьный файл *f*. Подсчитайте число вхождений в файл буквосочетаний «ab». Определите, входит ли в файл буквосочетание «abcdefgh». Подсчитайте число вхождений в файл каждой из букв «a», «b», «c», «d», «e», «f» и выведите результат в виде таблицы

a – N b – N c – N

$d - N$ $e - N$ $f - N$

3. Дан текстовый файл f . Получите самую длинную строку файла. Если в файле имеется несколько строк с наибольшей длиной, то получите одну из них.

Задача 9

1. Дан файл f , компоненты которого являются целыми числами. Ни одна из компонент файла не равна нулю. Файл f содержит столько же отрицательных чисел, сколько и положительных. Используя вспомогательный файл h , перепишите компоненты файла f в файл g так, чтобы в файле g :

а) не было двух соседних чисел с одним знаком;

б) сначала шли положительные, потом отрицательные числа;

в) числа шли в следующем порядке: два положительных, два отрицательных, два положительных, два отрицательных и т.д. (предполагается, что число компонент в файле f делится на 4).

2. Дан символьный файл f . Запишите в файл g компоненты файла f в обратном порядке.

3. Дан текстовый файл f . Исключите пробелы, стоящие в концах его строк. Результат поместите в файл f .

Задача 9

1. Дан файл f , компоненты которого являются целыми числами. Число компонент файла делится на 100. Запишите в файл g наибольшее значение первых ста компонент файла f , затем следующих ста компонент и т.д.

2. Даны символьные файлы f и g . Запишите в файл h все начальные совпадающие компоненты файлов f и g .

3. Даны текстовый файл и строка s . Получите все строки файла f , содержащие в качестве фрагмента строку s .

Задача 10

1. Дан файл f , компоненты которого являются целыми числами. Получите файл g , образованный из файла f исключением повторных вхождений одного и того же числа.

2. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Дан файл f , содержащий сведения о нескольких автомобилях. Найдите:

а) фамилии владельцев и номера автомобилей данной марки;

б) количество автомобилей каждой марки.

3. Даны два текстовых файла f и g . Определите, совпадают ли компоненты файла f с компонентами файла g . Если нет, то получите номер первой строки и позицию первого символа в этой строке, в которых файлы f и g отличаются между собой.

Тема 7. Несвязанные динамические структуры данных

1. Сведения о 10 персонах (дата рождения: число, день недели, месяц, год; место жительства: город, улица, номер дома, номер квартиры);
2. Семинар (предмет, преподаватель, номер группы, день недели, часы занятий, аудитория);
3. Сведения о компьютерах (название; объем памяти; быстродействие, стоимость в рублях и в долларах; фирма-поставщик; тип монитора и дисководов);
4. Бланк требования на книгу (сведения о книге: шифр, автор, название, год издания; сведения о читателе: номер читательского билета, фамилия; дата заказа);
5. Экзаменационная ведомость (предмет, номер группы, дата экзамена, 25 строчек с полями: фамилия студента, номер его зачетной книжки, оценка за экзамен);
6. Сведения о 10 студентах группы (фамилия, имя, отчество; дата рождения: число, месяц, год; пол; семейное положение; отношение к воинской службе);
7. Сведения о заводе-изготовителе компьютеров (название завода; название заводо-поставщиков комплектующих изделий; номера магазинов потребителей; название производимых компьютеров (5.. 10 наименований): цена каждого товара за штуку в рублях и в долларах; количество каждого вида изделий в штуках);
8. Сведения о 5 персонах (фамилия, имя, отчество, пол, национальность, семейное положение, образование (высшее, среднее специальное, среднее, неполное среднее); адрес: улица, номер дома, номер квартиры, почтовый индекс, номер телефона).
9. Сведения о заводе-изготовителе автомобилей (название завода, название заводо-поставщиков комплектующих изделий, номера фирменных магазинов, марка производимых автомобилей, цена автомобиля в рублях и в долларах, количество каждой выпускаемой марки автомобилей).
10. Сведения об успеваемости студентов (фамилия, имя, отчество, номер зачетной книжки, список оценок по предметам).

Тема 8. Связанные динамические структуры данных. Методы программирования с использованием связанных динамических структур данных

1. По списку L постройте два новых списка: L1 – из положительных элементов и L2 – из остальных элементов списка L (тип элементов – Real).
2. Найдите среднее арифметическое всех элементов непустого списка L (тип элементов – Real).
3. Объедините два упорядоченных по неубыванию списка L1 и L2 (тип элементов – Real) в один новый упорядоченный по неубыванию список L.
4. Включите в упорядоченный список элемент, сохраняя упорядоченность.
5. Проверьте на равенство две очереди. Решение в программе оформите через подпрограммы.
6. Найдите все вершины заданного графа, недостижимые от заданной его вершины.
7. Для двух выделенных вершин графа постройте соединяющий их простой путь.
8. Задана система односторонних дорог. Найдите путь, соединяющий города A и B и не проходящий через заданное множество городов.
9. Задана система двусторонних дорог. Найдите два города и соединяющий их путь, который проходит через каждую из дорог системы ровно один раз.
10. Задана система двусторонних дорог, причем для любой пары городов можно указать соединяющий их путь. Найдите такой город, для которого сумма расстояний до остальных городов минимальна.

Тема 9. Понятие и методы объектно-ориентированного программирования

1. Student: Фамилия, Имя, Отчество, Дата рождения, Адрес, Телефон, Факультет, Курс.

Создать массив объектов. Вывести:

- а) список студентов заданного факультета;
- б) списки студентов для каждого факультета и курса;
- в) список студентов, родившихся после заданного года.

2. Abiturient: Фамилия, Имя, Отчество, Адрес, Оценки. Создать массив объектов. Вывести:

- а) список абитуриентов, имеющих неудовлетворительные оценки;
- б) список абитуриентов, сумма баллов у которых не меньше заданной;
- в) выбрать N абитуриентов, имеющих самую высокую сумму баллов, и список абитуриентов, имеющих полупроходной балл.

3. Aeroflot: Пункт назначения, Номер рейса, Тип самолета, Время вылета, Дни недели. Создать массив объектов. Вывести:

- а) список рейсов для заданного пункта назначения;
- б) список рейсов для заданного дня недели;
- в) список рейсов для заданного дня недели, время вылета для которых больше заданного.

4. Book: Автор, Название, Издательство, Год, Количество страниц.

Создать массив объектов. Вывести:

- а) список книг заданного автора;
- б) список книг, выпущенных заданным издательством;
- в) список книг, выпущенных после заданного года.

5. Worker: Фамилия и инициалы, Должность, Год поступления на работу, Зарплата. Создать массив объектов. Вывести:

- а) список работников, стаж работы которых на данном предприятии превышает заданное число лет;
- б) список работников, зарплата которых больше заданной;
- в) список работников, занимающих заданную должность.

6. Train: Пункт назначения, Номер поезда, Время отправления, Число общих мест, Купейных, Плацкартных. Создать массив объектов. Вывести:

- а) список поездов, следующих до заданного пункта назначения;
- б) список поездов, следующих до заданного пункта назначения и отправляющихся после заданного часа;
- в) список поездов, отправляющихся до заданного пункта назначения и имеющих общие места.

7. Product: Наименование, Производитель, Цена, Срок хранения, Количество. Создать массив объектов. Вывести:

- а) список товаров для заданного наименования;
- б) список товаров для заданного наименования, цена которых не превышает указанной;
- в) список товаров, срок хранения которых больше заданного.

8. Patient: Фамилия, Имя, Отчество, Адрес, Номер медицинской карты, Диагноз. Создать массив объектов. Вывести:

- а) список пациентов, имеющих данный диагноз;
- б) список пациентов, номер медицинской карты которых находится в заданном интервале.

9. Bus: Фамилия и инициалы водителя, Номер автобуса, Номер маршрута, Марка, Год начала эксплуатации, Пробег. Создать массив объектов. Вывести:

- а) список автобусов для заданного номера маршрута;
- б) список автобусов, которые эксплуатируются больше 10 лет;
- в) список автобусов, пробег у которых больше 10 000 км.

10. Customer: Фамилия, Имя, Отчество, Адрес, Телефон, Номер кредитной карточки, Номер банковского счета. Создать массив объектов. Вывести:

- а) список покупателей в алфавитном порядке;
- б) список покупателей, номер кредитной карточки которых находится в заданном интервале.

Тема 10. Операторный базис языка программирования C++

1. Вычислите длину окружности, площадь круга и объём шара одного и того же заданного радиуса.
2. Определите номер квадранта, в котором находится точка с заданными координатами (x, y) .
3. Подсчитайте число и сумму положительных, число и произведение отрицательных элементов заданного массива $A(N)$.
4. Запишите подряд в массив $A(N)$ элементы заданного массива $B(2N)$, стоящие на чётных местах, а элементы, стоящие на нечетных местах, запишите в массив $C(N)$.
5. Дана матрица $A(N, M)$. Найдите её наибольший элемент и номера строки и столбца, на пересечении которых он находится.
6. Дана матрица $A(N, N)$ и целое P . Преобразуйте матрицу по правилу: строку с номером P сделайте столбцом с номером P , а столбец с номером P сделайте строкой с номером P .
7. Выясните, имеются ли в заданном векторе $A(N)$ два подряд идущих нулевых элемента.
8. Все элементы заданного вектора $A(N)$, начиная с первого по порядку положительного элемента, уменьшите на единицу.
9. В заданной целочисленной матрице $A(N, M)$ выведите на печать индексы первого положительного элемента, кратного заданному числу K . Если таких элементов в матрице нет, то выведите соответствующий текст. Элементы матриц просматривайте слева направо и сверху вниз.
10. Элементы заданной матрицы $A(N, N)$ переписывайте построчно в одномерный массив до тех пор, пока не встретится нулевой элемент.
11. В заданной матрице $A(N, M)$ найдите количество строк, не содержащих отрицательных чисел.
12. Дана матрица $A(N, M)$. Постройте вектор $B(N)$, элементы B_i которого равны единице, если элементы i -й строки образуют упорядоченную по убыванию или по возрастанию последовательность, и нулю во всех остальных случаях.

Тема 11. Методы трансляции

1. Постройте синтаксический анализатор для понятия *простое-выражение*:
простое выражение ::=

$$\left\{ \begin{array}{l} \text{простой-идентификатор} \\ \text{простое-выражение} \text{ знак-операции } \text{простое-выражение} \end{array} \right\}$$

простой идентификатор ::= буква

$$\text{знак операции} ::= \left\{ \begin{array}{c} - \\ + \\ * \end{array} \right\}$$

2. Постройте синтаксический анализатор для понятия *идентификатор*:

$$\text{идентификатор} ::= \left\{ \begin{array}{l} \text{буква} \\ \text{идентификатор} \left\{ \begin{array}{l} \text{цифра} \\ \text{буква} \end{array} \right\} \end{array} \right\}$$

3. Постройте синтаксический анализатор для понятия *вещественное число*:

вещественное число ::=

$$\left\{ \begin{array}{l} \text{целое-число} \text{целое-без-знака} \\ \text{целое-число} \text{целое-без-знака} E \text{целое-число} \\ \text{целое-число} E \text{целое-число} \end{array} \right\}$$

целое-без-знака ::= *цифра* {*цифра*}*

$$\text{целое-число} ::= \left\{ \begin{array}{l} \text{целое-без-знака} \\ \left\{ \begin{array}{l} + \\ - \end{array} \right\} \text{целое-без-знака} \end{array} \right\}$$

4. Постройте синтаксический анализатор для понятия *константное выражение*:

константное-выражение ::=

$$\left\{ \begin{array}{l} \text{цифра} \{ \text{цифра} \}^* \\ \text{константное-выражение} \left\{ \begin{array}{c} - \\ + \\ * \end{array} \right\} \text{константное-выражение} \end{array} \right\}$$

5. Постройте синтаксический анализатор для понятия *простое-логическое*:

простое-логическое ::=

$$\left\{ \begin{array}{l} TRUE \\ FALSE \\ \text{простой-идентификатор} \\ NOT \text{простое-логическое} \\ (\text{простое-логическое} \text{ знак-операции} \text{ простое-логическое}) \end{array} \right\}$$

простой-идентификатор ::= *буква*

$$\text{знак-операции} ::= \left\{ \begin{array}{l} AND \\ OR \end{array} \right\}$$

Критерии формирования оценок по контрольным точкам (коллоквиум)

«отличный (высокий) уровень компетенции» (5 баллов) - ставится в случае, когда обучающийся демонстрирует знание теоретического материала на 100%;

«хороший (нормальный) уровень компетенции» (4 баллов) - ставится в случае, когда обучающийся демонстрирует знание теоретического материала на 70%;

«удовлетворительный (минимальный, пороговый) уровень компетенции» (3 балла) – ставится в случае, когда обучающийся затрудняется с правильной формулировкой теоретического материала, дает неполный ответ, демонстрирует знание теоретического материала на 50%;

«неудовлетворительный (ниже порогового) уровень компетенции» (2 и менее баллов) –

ставится в случае, когда обучающийся дает неверную формулировкой теоретического материала, дает неверный ответ, демонстрирует незнание теоретического материала или знание материала менее чем на 40%.

Оценочные материалы для контрольной работы, (контролируемая компетенция ПКС-3)

1. Какие основные области применения языка Си++?

- + (1) системное программирование
- + (2) прикладное программирование
- (3) программирование дизайна сайтов

2. В чем заключается суть компоновки программы?

- (1) в переводе текстового файла в объектный модуль
- (2) в подготовке программы к выполнению
- + (3) в объединении нескольких фрагментов программы в один

3. У какой переменной в данном коде самое короткое «время жизни»? `char foo(char my_ch) { char ch= my_ch; static int flag = 1; if (flag){ char p; p=ch; ch=ch+1; } return ch; }`

- + (1) p
- (2) flag
- (3) ch

4. Какие ключевые слова используются для создания и обработки исключительных ситуаций?

- + (1) try
- (2) delete
- + (3) catch
- (4) return
- + (5) throw

5. Что понимается под потоком в языке С++

- (1) обмен данными между программами
- (2) обмен данными между компилятором и функцией main
- + (3) механизм ввода-вывода

6. Укажите неправильный идентификатор:

- (1) AB_D1
- + (2) 10xd
- (3) z1d8_14f3

7. Отметьте фрагменты кода, которые можно назвать выражениями:

+ (1) $f + r * 12 - 14$

- (2) `int z;`

+ (3) `x = y = 13`

8. Укажите правильное объявление?

- (1) `int 5;`

- (2) `float fl, int i`

+ (3) `float F3v7G8t9F; int iCr3;`

9. Прототип функции задает

+ (1) тип функции, включая количество и тип аргументов и тип результата

- (2) возможность выполнения этой функции из программ на других языках программирования

- (3) имя функции и минимальное количество параметров

10. Какие из перечисленных типов являются встроенными типами языка C++?

+ (1) `float`

- (2) `real`

- (3) `integer`

+ (4) `bool`

11. Если в массиве A 132 элемента, каким будет правильное обращение к последнему элементу массива?

- (1) `A[132]`

+ (2) `A[131]`

- (3) `A[133]`

- (4) `A[-1]`

12. Что из себя представляет динамическое выделение памяти?

- (1) память под объект (переменную) выделяется каждый раз при обращении к переменной

- (2) память под объект (переменную) может выделяться не сразу, а в процессе работы программы, освобождение памяти производится автоматически после завершения программы

+ (3) память под объект (переменную) может выделяться не сразу, а в процессе работы программы, освобождение памяти производится вручную

13. Программа на языке Си++ начинает выполняться с:

- (1) первой функции в программе
- + (2) функции main
- (3) той функции, которая указана как стартовая при компиляции программы

14. Какой длины может быть идентификатор

- (1) не более 64 символов
- (2) не более 128 символов
- + (3) в самом языке по длине нет ограничений

15. Отметьте, какому определению функции может соответствовать вызов func(5.98):

- (1) `int func(double x = 0, double y);`
- + (2) `void func(double x);`
- + (3) `double func(double x, int y = 12, int z = 5);`
- + (4) `void func(float arg1, float arg2 = 0);`
- (5) `float func(float g, float f);`

16. Компилятор языка Си++:

- + (1) переводит текст программы в машинные инструкции
- (2) выполняет программу
- (3) форматирует текст программы так, чтобы его было удобно читать

16. Какое слово из списка не относится к зарезервированным словам Си++?

- (1) try
- (2) union
- + (3) cast
- (4) volatile

17. Если функция вычисления суммы целых чисел от 1 до n имеет прототип `int`

`sum(int n)`, запишите определение функции, используя рекурсию:

- + (1) `{ if (n == 1) return 1; else return n + sum (n - 1); }`
- (2) `{ if (n == 1) return 1; else return sum(n); }`
- (3) `{ if (n == 1) return 1; else return sum(n) + (n - 1); }`
- (4) `{ if (n == 1) return 1; else return sum(n) + sum (n-1); }`

18. Что произойдет после объявления в программе данного набора перечисляемых значений: `enum{N=0, E=1, S=2, W=3};`?

- (1) программа будет работать с числовыми значениями N, E, S и W

- + (2) программа будет работать с идентификаторами N, E, S и W
- (3) программа сформирует итераторы с указателями на N, E, S и W
- (4) программа будет игнорировать все указатели на N, E, S и W

19. Процесс компиляции программы

- + (1) переводит исходный текст в исполняемый файл
- (2) приводит программы к единообразному внешнему виду
- (3) для языка Си++ необязателен

- + (1) `void func(const Foo& a);`
- + (2) `void func(Foo& a) const;`
- + (3) `void func(const Foo& a) const;`

20. Объявление `extern int f;` означает:

- + (1) переменная определена в другом исходном файле
- (2) переменная определена в стандартной библиотеке
- (3) переменная может быть использована только в этом файле

21. Выберите правильное объявление константы `pi`:

- + (1) `const float pi = 3.14;`
- (2) `float pi = (const) 3.14;`
- (3) `const float pi 3.14;`

22. Операция `++`

- + (1) увеличивает значение переменной на единицу
- (2) увеличивает значение переменной на два
- (3) уменьшает значение переменной на единицу
- (4) уменьшает значение переменной на два
- (5) в языке Си++ не существует

23. Что означает запись `for (;;)?`

- + (1) бесконечный цикл
- (2) цикл, который не выполняется ни разу
- (3) ошибка компиляции
- (4) аварийный выход из программы

24. Совокупность типов формальных параметров, их порядка и имени функции

определяет:

- (1) тип возвращаемого функцией значения
- + (2) сигнатуру (подпись) функции
- (3) идентификатор функции
- (4) последовательность описаний и определений функции

25. Для получения адреса переменной используется операция

- (1) *
- + (2) &
- (3) ->
- (4) .
- (5) нет правильного ответа

26. В программе на языке Си++ обязательно имеется функция

- (1) head
- (2) start
- (3) prime
- + (4) main
- (5) finish

27. Отметьте правильные объявления переменных

- + (1) `const int s = 10; int a[s];`
- (2) `int s = 10; const int a[s];`
- (3) `int s = 10; int a[] = new int[s+s];`

28. Отметьте, какие возможности языка Си++ помогают предупреждать ошибки:

- (1) наличие встроенных типов данных
- + (2) контроль типов при компиляции
- (3) возможность использовать указатели вместо массивов
- + (4) обязательность объявления функций до их использования

29. Какой класс используется для вывода данных во внутреннюю область памяти?

- (1) `iostream`
- + (2) `stringstream`
- (3) `cout`
- (4) `strout`
- (5) `fstream`

30. Отметьте правильное определение константы:

- + (1) `const int DOZEN = 12;`
- (2) `const double;`
- (3) `double COFF = (const)1.2e10;`

31. Укажите в каких выражениях правильно определены целочисленные переменные?

- + (1) `short x = 23;`
- + (2) `unsigned int a = 2, b = 3;`
- (3) `const unsigned char c;`
- + (4) `unsigned short g = 0x2;`

32. Ключевое слово `void` обозначает что функция

- (1) возвращает число с плавающей запятой
- (2) возвращает целое число
- + (3) ничего не возвращает
- (4) является главной

33. В каких строках ошибка? `new long[];` `new long[][2][4];` `new long[3][][4];`

- + (1) во всех строках ошибка – неизвестен размер
- (2) в строках нет ошибки
- (3) в первой строке ошибка – неизвестен размер
- (4) в последней строке ошибка – неправильный синтаксис
- (5) во всех строках ошибка – неправильный синтаксис

34. Какой тип преобразования типов используется в следующем выражении? `int a =`

`0; float f = 3.4; a += f;`

- + (1) неявное преобразование типов
- (2) явное преобразование типов
- (3) пользовательское преобразование типов

35. С помощью какой директивы происходит подключение других модулей программы?

- + (1) `#include`
- (2) `#define`
- (3) `#ifndef`

36. Что является минимальной областью видимости имен?

- (1) модуль
- + (2) блок
- (3) функция
- (4) класс

37. В каких выражениях используются бинарные арифметические операции?

- + (1) $X + Y$
- + (2) $c \% d + 2$
- + (3) $xx * Y$
- (4) $xx++$

38. Какое из следующих утверждений об операторе return является верным?

- (1) оператор return должен стоять последним в теле функции
- + (2) оператор return завершает выполнение функции
- (3) в теле функции должен присутствовать только один оператор return

39. Что описывает данная строка программы: float mas=new int[3][2]?

- (1) создание двумерного динамического массива размерности 3*2
- (2) создание одномерного динамического массива из 3 элементов
- (3) создание одномерного динамического массива из 2 элементов
- + (4) данная строка представляет собой ошибочную запись и работать не будет

40. Какие компоненты могут входить в интегрированную среду программирования

- + (1) текстовый редактор
- + (2) отладчик
- + (3) компилятор

41. Для чего предназначены манипуляторы потоков ввода-вывода?

- + (1) для управления состоянием потока
- (2) для перенаправления потоков ввода-вывода
- (3) для создания объектов классов istream и ostream

42. Укажите правильные присваивания значений переменным и константам

- (1) `float Y = 12, int Y = 3;`
- + (2) `int iCode = 12123; int Viv_12 = iCode;`
- + (3) `const int Const = 2; int iConst = Const + 3;`

43. Укажите в каком выражении используется операция с наивысшим приоритетом?

- (1) $x \ll 3$
- (2) $c + D$
- (3) $a2 \gg 5$
- + (4) $k++$

44. Какое определение функции является правильным?

- (1) `int f(int b,int c) {return;}`
- (2) `void f(int b,int c) {return b+c;}`
- + (3) `int f(int b,int c) {return 0;}`

45. Представление и диапазоны значений вещественных чисел соответствуют

стандарту:

- (1) ASCII
- (2) ANSI
- + (3) IEEE
- (4) CP-1251

46. Для создания исполняемого файла в системе Unix необходимо

- (1) только набрать текст в текстовый файл
- (2) только запустить любой из имеющихся компиляторов: GNU C++, g++, c++, cc
- + (3) набрать текст в текстовый файл и запустить его на компиляцию любым из имеющихся компиляторов C++

47. Если в арифметическом выражении участвуют целый и вещественный

операнды, то:

- + (1) целый тип приводится к вещественному
- (2) вещественный тип приводится к целому
- (3) ошибка компиляции

48. Отметьте истинные высказывания в отношении использования макросов и

функций:

- + (1) в большинстве случаев функции позволяют сокращать объем выполняемого файла
- + (2) в большинстве случаев макросы позволяют сокращать время выполнения
- + (3) недостатком макросов является отсутствие встроенного контроля согласования типов аргументов и формальных параметров

49. Если имеется объявление `char ch1='a',ch2='b',ch3='c'`; допустима ли

запись `ch1=ch2+ch3`;

-(1) нет

+(2) да

50. Какие операторы не могут быть переопределены пользователем:

-(1) %

+(2) ::

+(3) .

51. Существует ли в C++ готовый набор шаблонов:

-(1) нет

+(2) да, существует специальная библиотека STL

-(3) зависит от версии компилятора

52. Какой статус международного стандарта языка Си++?

-(1) принят только в США и ждет одобрения международной организации

+(2) принят ISO и тем самым автоматически принят во всех странах

-(3) принят проект стандарта, дорабатывается

53. Что является результатом компоновки программы?

-(1) заголовочный файл

+(2) исполняемый файл или библиотека

-(3) набор заголовочных файлов с определением в них всех используемых функций

54. Функция вычисляет произведение двух чисел. Исходные данные вводятся с клавиатуры. Какие проверки целесообразно ввести в программе?

-(1) проверка исходных данных на равенство нулю

-(2) проверка, что исходные данные являются числами и эти числа больше нуля

+(3) проверка, что исходные данные являются числами

-(4) проверки не нужны, все возможные ошибки отловит компилятор

55. Укажите правильный идентификатор для имени переменной:

-(1) FA_Ф12

+(2) _ri18

-(3) int

-(4) 2a

56. Операции в выражениях могут быть

- (1) только унарными или бинарными
- (2) только бинарными
- +(3) унарными, бинарными или тернарными

57. Сколько функций может быть в программе C++?

- (1) ни одной
- +(2) минимум одна
- (3) не больше 100

58. Какие из следующих выражений являются константами типа double?

- +(1) 3.0 + 2
- +(2) 5.00E+06
- +(3) 0.8
- (4) 12IU

59. Если имеется объявление `int a[15]`, какой фрагмент кода выводит все элементы массива?

- (1) `for (int k = 0; k <= 15; k++) cout << a[k];`
- (2) `for (int k = 1; k < 16; k++) cout << a[k];`
- +(3) `for (int k = 0; k < 15; k++) cout << a[k];`

60. Какое из приведенных выражений верно?

- (1) `int a; a = new int[20];`
- (2) `int a; a = new int(20);`
- +(3) `int *a; a = new int[20];`
- (4) `int *a; a = new 20;`
- (5) `int *a; a = new sizeof(int*20);`
- (6) `int a; a = new sizeof(int*20);`

61. Что такое `cout`?

- +(1) объект типа `iostream` (`std::ostream`)
- (2) класс, который выводит данные на терминал
- (3) переменная, которую программист должен создать для вывода данных

62. Сколько вызовов системных функций всегда будет в программе, независимо от ее реализации?

- (1) всегда 2
- + (2) по крайней мере один
- (3) не менее двух

63. На каком этапе обнаруживаются ошибки в алгоритме программы?

- (1) на этапе компиляции
- + (2) на этапе выполнения
- + (3) они могут не проявиться никогда, все зависит от входных данных

64. Если есть два объявления `int qwerty; int QWERTY;` какое из утверждений верно

- (1) такие имена переменных недопустимы
- + (2) объявления правильные
- (3) такие объявления недопустимы, так как мы пытаемся создать две одинаковые переменные

65. Если в программе уже имеется функция с прототипом `int func(int k, double f)`, то какое из следующих объявлений не вызовет ошибки компиляции?

- (1) `double func(int m, double g)`
- + (2) `int func(double x, int y)`
- (3) `void func(int m, double g = 3.14)`

66. Определите размер структуры

```
struct { char fio[30]; unsigned char date:4; unsigned char code:4; };
```

- (1) 9 байт
- (2) 38 байт
- + (3) 31 байт
- (4) 39 байт

67. После компиляции программы

- + (1) ее можно выполнять многократно без перекомпиляции
- (2) перед каждым последующим запуском ее нужно перекомпилировать
- (3) ее можно выполнять только с одним набором исходных данных

68. Какой из стандартных классов используется для вывода строк на терминал:

- (1) `stringstream`
- + (2) `ostream`
- (3) `ofstream`
- (4) `istream`

(5) ifstream

69. Какое из приведенных имен является недопустимым в Си++?

-(1) x03488erJJJ_____

+(2) xb____@

-(3) r13

-(4) OOP

70. Какой из наборов перечисляемых значений записан правильно?

-(1) enum { a, b, 3, 4 };

-(2) enum { a, b = 3, c = 4, 3 };

+(3) enum { a, b = 3, c, d };

71. Комментарий в программе на Си++

-(1) содержит указания компилятору по настройке программы

+(2) содержит пояснения к тексту и не оказывает влияния на выполнение программы

-(3) должен содержать допустимые аргументы программы

72. Могут ли контексты быть вложенными?

+(1) могут

-(2) не могут

-(3) могут, при определенных условиях

73. Является ли x переменной или константой? $y = 12 + x$;

-(1) является переменной

-(2) является константой

+(3) определить нельзя

74. Нелогической операцией является

-(1) &&

-(2) ==

-(3) ||

-(4) !

+(5) =

75. Что означает запись while (false);?

-(1) бесконечный цикл

+(2) цикл, который не выполняется ни разу

-(3) ошибка компиляции

-(4) аварийный выход из программы

76. Если функция имеет тип void, то неверно, что

+(1) она не может содержать оператор return

-(2) оператор return в теле функции необязателен

-(3) функция не возвращает никакого значения

77. Какая операция позволяет получить значение, записанное по адресу, который содержится в указателе?

- + (1) *
- (2) ?
- (3) ^
- (4) &
- (5) .

78. Что произойдет при использовании неправильного адреса в операции delete?

- (1) произойдет аварийное завершение программы
- (2) программа выдаст сообщение, что память освобождается по неправильному адресу
- + (3) результат непредсказуем

79. При выходе из функции main

- (1) программа повторяется с теми же аргументами
- + (2) программа заканчивается
- (3) выполняется функция finish, определенная программистом

80. Какое из следующих объявлений является объявлением неизменяемого указателя?

- (1) `int const* ptr;`
- (2) `const int* ptr;`
- (3) `int * ptr const;`
- + (4) `int* const ptr;`

81. Какова последовательность создания исполняемого файла:

- Создать файл с исходным текстом программы, который будет иметь расширение `.c`.
- Скомпоновать объектный файл с необходимыми библиотеками.
- Скомпилировать исходный код.

- + (1) 1-3-2
- (2) 2-3-1
- (3) 1-2-3
- (4) 3-2-1
- (5) 2-1-3

82. С помощью какого метода можно изменить текущую позицию в файле?

- (1) `put`
- (2) `open`
- + (3) `seekp`

83. Отметьте все неправильные определения констант:

- + (1) `const int 12X = 12;`

- (2) `const int DAYS_OF_WEEK = 7;`
- + (3) `int const x = 2i+56;`

84. Как называется функция, которая вызывает саму себя?

- (1) конструктором
- (2) деструктором
- (3) подставляемой
- + (4) рекурсивной

85. Укажите в каком выражении правильно определена переменная в шестнадцатеричной системе счисления?

- + (1) `short x = 0x1244;`
- (2) `char c = 9340x;`
- (3) `unsigned long l = 0x84GAF;`
- + (4) `int k = 0xCDeF;`

86. Отметьте правильный вариант освобождения всей памяти, выделенной для трехмерного массива для следующей программы `long (*lp)[2][4]; lp = new`

`long[3][2][4];`

- + (1) `delete [] lp;`
- (2) `delete lp;`
- (3) `delete [][] lp;`
- (4) `delete [][] lp;`

87. Какое приведение типов используется в следующем выражении? `int a = 0; float f =`

`3.4; f += (int)a;`

- (1) неявное приведение типов
- + (2) явное приведение типов
- (3) стандартное приведение типов

88. В чем различие использования следующих выражений

`#include <...>` и `#include "..."`

- (1) нет различий
- + (2) различие заключается в методе поиска препроцессором включаемого файла
- (3) в различии использования заголовочных и исходных файлов

89. Для чего предназначен оператор `namespace`?

- + (1) для заключения в группу объявлений классов, переменных и функций в отдельный контекст со своим именем
- (2) для заключения в группу объявлений классов, переменных и функций для использования только в текущем модуле
- (3) для использования классов, переменных и функций из других модулей программы без использования заголовочных файлов

90. Что означает `cout << setw(3)` ?

- + (1) ширина поля вывода устанавливается равной 3
- (2) выводимые строки сокращаются до 3 символов
- (3) выводимые строки дополняются до 3 символов
- (4) нельзя ввести больше 3 символов за один раз

91. В каких выражениях используются унарные арифметические операции?

- (1) `c1 + d2`
- (2) `s2 % d % 2`
- + (3) `--b`
- + (4) `d++`

92. Файл имеющий имя `"test_file.cpp"` это:

- (1) исполняемый файл
- (2) заголовочный файл
- + (3) файл исходного текста языка C++
- (4) динамически загружаемая библиотека

93. Укажите основные используемые манипуляторы потоков.

- + (1) `flush`
- + (2) `endl`
- (3) `printf`
- + (4) `setw`
- (5) `<<`
- + (6) `setprecision`

94. Укажите, где происходит объявление констант или переменных

- + (1) `float dD;`
- + (2) `float sd2 = 3.2;`
- (3) `bType = 3;`
- + (4) `int k; k = 89;`

95. В каких выражениях возвращаемое значение будет логическим?

- (1) `int x = 2, y = 3, z; z = x + y;`

- + (2) $(i > 3)$;
- + (3) $(c == b)$;

96. В чем заключается назначение оператора перехода goto?

- + (1) изменяет последовательность выполнения операторов в программе
- (2) необходим для передачи аргумента в функцию
- (3) для наглядного отображения листинга программы
- (4) для перехода из одной функции в другую

97. Укажите в каком выражении произойдет потеря точности

- + (1) `int i; float x = 2.134, y = 3.14; i = x/y;`
- (2) `float M = 235.2; double Z = 3; Z *= M;`
- (3) `short i = 0x3; float x = 2.7, v; v = i + x;`

98. Укажите какие компиляторы языка Си++ могут быть использованы в системе

Unix.

- (1) Microsoft© Visual C++
- (2) Emacs
- + (3) GNU C++, g++, c++
- (4) любой компилятор языка C++

99. Если в арифметическом выражении участвуют короткое целое и длинное целое,

то:

- (1) длинное приводится к короткому
- + (2) короткое приводится к длинному
- (3) ошибка компиляции

100. Отметьте истинные высказывания:

- + (1) переменная объявляется, потом изменяется
- (2) переменная инициализируется, потом объявляется
- (3) переменная объявляется, потом инициализируется и изменяется

Критерии формирования оценок по тестовым заданиям:

По итогам выполнения тестовых заданий оценка производится по пятибалльной шкале. При правильных ответах на:

- 89-100% заданий – «5» (баллов);
- 70-88% заданий – «4» (баллов);
- 50-69% заданий – «3» (балла);
- 30-49% заданий – «2» (балла);
- 10-29% заданий – «1» (балл);
- менее 10% заданий – «0» (баллов).

Полный перечень вопросов, выносимых на экзамен (контролируемая компетенция ПКС-3)

№	Вопрос	Код компетенции (согласно РПД)
1.	Архитектура веб-приложений. Протокол HTTP. Браузеры. Веб-серверы.	ПКС-3
2.	Основные языки программирования. W3C. Языки разметки SGML, HTML, XML, RSS, KML. CSS.	ПКС-3
3.	Статические и динамические веб-страницы. Программирование на Javascript. W3C DOM. Средства отладки.	ПКС-3
4.	Библиотеки jQuery, Dojo	ПКС-3
5.	Универсальные скриптовые языки программирования. Основы Ruby. Базовые конструкции. Блоки. Массивы.	ПКС-3
6.	Основы Ruby. Классы. Модули. Метaprogramмирование.	
7.	Ruby. Функциональный стиль программирования.	ПКС-3
8.	Принципы работы веб-серверов. Принципы построения веб-приложений. CGI. Фильтры и обработчики запросов. Переменные окружения. Передача параметров и файлов серверным модулям приложения.	ПКС-3
9.	Создание веб-приложений в форме CGI и в форме шаблона со вставками. Ruby CGI, erb.	ПКС-3
10.	Принципы построения веб-приложений с использованием Ruby. Rack, Sinatra, Ruby on Rails.	ПКС-3
11.	Шаблоны проектирования Ruby on Rails. Схема Model-View-Controller.	ПКС-3
12.	Написание тестов веб-приложений. Unit test, RSpec, Selenium webdriver, тесты RoR.	ПКС-3
13.	Синхронный и асинхронный (AJAX) принципы построения веб-интерфейса. Языки и форматы обмена данными.	ПКС-3
14.	Хранение состояния на стороне браузера и на стороне сервера. Cookies, WebStorage, Sessions.	ПКС-3
15.	Способы обращения к базам данных. ORM (Object-relational mapping) в Ruby on Rails.	ПКС-3
16.	Проблемы безопасности веб-приложений.	ПКС-3
17.	Языки запросов XPath, XQuery.	ПКС-3
18.	Языки преобразования данных XSL, XQuery.	ПКС-3
19.	Принципы работы веб-серверов. Принципы построения веб-приложений. CGI. Фильтры и обработчики запросов. Переменные окружения. Передача параметров и файлов серверным модулям приложения.	ПКС-3

20.	Веб-сервисы. Спецификация WSDL. Протоколы SOAP и REST. Средства для работы с сервисами.	ПКС-3
21.	Веб-серверы Apache, nginx, jBoss. Размещение приложений в сети Интернет. Управление серверами.	ПКС-3
22.	CMS (Content Management System) на примере Drupal, WordPress, Radiant, Octopress.	ПКС-3

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
**Федеральное государственное бюджетное образовательное учреждение высшего
образования «Кабардино-Балкарский государственный университет
им. Х.М. Бербекова» (КБГУ)**

Кафедра– Прикладной математики и информатики

Дисциплина – Языки и методы программирования

Направление подготовки – 02.03.02 Фундаментальная информатика и информационные технологии, 1 курс

Экзаменационный билет №1

- 1.
- 2.

Руководитель ОПОП _____ / _____ /

Зав. кафедрой ПМ и И _____ / _____ /