

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

для проведения текущей и промежуточной аттестации

по учебной дисциплине (модулю)

«Программирование»

для направления подготовки 44.03.05 Педагогическое образование
(с двумя профилями подготовки)
Профиль "Информатика и физика"

1. Описание показателей (дескрипторов) и критериев оценивания компетенций на различных этапах их формирования

Контроль качества освоения дисциплины включает в себя текущий контроль успеваемости и промежуточную аттестацию. Текущий контроль успеваемости и промежуточная аттестация обучающихся проводятся в целях установления соответствия достижений обучающихся поэтапным требованиям образовательной программы к результатам обучения и формирования компетенций.

Компетенции	Показатели* (дескрипторы)	Критерии в соответствии с уровнем освоения ОП			Оценочное средство (промежуточная аттестация)
		пороговый (удовлетворительно) 55-69 баллов	стандартный (хорошо) 70-84 балла	эталонный (отлично) 85-100 баллов	
ОПК-8	Знать	основные методы использования современных информационно-коммуникационных технологий для решения учебных задач.	основные методы использования современных информационно-коммуникационных технологий для решения практических задач.	основные методы использования современных информационно-коммуникационных технологий для решения профессиональных задач.	Диктант, доклад
	Уметь	использовать возможности информационных технологий для решения задач; оценивать собственные образовательные достижения, определять потребности в дальнейшем образовании.	использовать возможности информационных технологий для решения задач, самообразования; оценивать собственные образовательные достижения и проблемы, определять потребности в дальнейшем образовании.	использовать возможности информационных технологий для решения задач, самообразования; нести ответственность за результаты своих действий и качество выполненных заданий; оценивать собственные образовательные достижения и проблемы, определять потребности в дальнейшем образовании.	Лабораторная работа, домашняя работа
	Владеть	умением быть готовым к работе в команде, выполнению проектной деятельности.	навыками проведения проектной работы в рамках учебной информации.	навыками проведения научного исследования, проектной работы в профессиональной области.	Контрольная работа
ПК-1	Знать	базовые термины программирования на языках высокого уровня	терминологическую систему программирования на языках высокого уровня	основные алгоритмические структуры и методы программирования.	Диктант, доклад
	Уметь	использовать теоретические знания по программированию для разработки программного решения базовых учебных задач	использовать теоретические знания по программированию для разработки программного решения базовых задач	использовать теоретические знания по программированию для разработки программного решения задач	Лабораторная работа, домашняя работа

	Владеть	навыками использования полученных теоретических и практических знаний для решения учебных задач.	навыками использования полученных теоретических и практических знаний для решения базовых задач.	навыками использования полученных теоретических и практических знаний для решения задач.	Контроль ная работа
--	---------	--	--	--	------------------------

*Показатели (дескрипторы) перечисляются по всей компетенции, если индикаторы компетенции сформулированы в виде «действия».

2. Описание критериев и шкал оценивания результатов обучения по дисциплине

2.1. Критерии и шкалы оценивания результатов обучения при проведении текущего контроля успеваемости

Текущий контроль предназначен для проверки хода и качества формирования компетенций, стимулирования учебной работы обучающихся и совершенствования методики освоения новых знаний. Он обеспечивается проведением семинаров, оцениванием контрольных заданий, проверкой конспектов лекций, выполнением индивидуальных и творческих заданий, периодическим опросом обучающихся на занятиях. Контролируемые разделы (темы) дисциплины, компетенции и оценочные средства представлены в таблице.

4 семестр

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции и/или индикаторы компетенции	Наименование оценочного средства**
1	Характеристики структурированных типов данных. Массивы. Линейные и двумерные массивы. Длинная арифметика. Строки. Множества. Записи. Типизированные файлы. Организация файлов записей. Нетипизированные файлы. Текстовые файлы. Прямой доступ к компонентам файлов. • Сортировка файлов.	ОПК-8 ПК-1	Лабораторная работа Домашняя работа Выполнение контрольной работы №1
2	Процедуры. Разработка и вызов. Функции. Разработка и вызов. Разработка программ на основе структурного подхода. Внешние подпрограммы. Рекурсивные подпрограммы. Модули. Структура и разработка	ОПК-8 ПК-1	Лабораторная работа Домашняя работа Итоговый контрольный тест 4 семестра Расчетное задание Итоговое собесед. Выполнение индивидуальных заданий

5 семестр

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции и/или индикаторы компетенции	Наименование оценочного средства**
3	Методология разработки объектноориентированного программного обеспечения. Объектноориентированный анализ и проектирование; Абстрактные типы и структуры данных; Классы, объекты, поля, методы; Конструкторы и деструкторы; Свойства и методы объектов; Раннее связывание и позднее связывание; Математические объекты: рациональные и комплексные числа, вектора, матрицы.	ОПК-8 ПК-1	Лабораторная работа Домашняя работа Индивидуальное задание №1
4	Стандартные модули. Динамически распределяемая память и ее использование при работе со стандартными типами данных. Однонаправленные списки. Двухнаправленные списки. Стеки. Очереди. Деки. Двоичные деревья поиска.	ОПК-8 ПК-1	Лабораторная работа Домашняя работа Расчётное графическое задание Итоговое собеседование Проект

Критерии и шкала оценивания лабораторных и домашних работ

Правильно выполненная лабораторная работа	1 балл
Правильно выполненная домашняя работа	0,5 балла

Критерии и шкала оценивания тестирования

Оценка	Критерий оценки
«зачтено»	Выполнение более 60% тестовых заданий 12 баллов
«не зачтено»	Выполнение менее 60% тестовых заданий

Критерии и шкала оценивания итогового собеседования обучающихся на занятии

Итоговое собеседование проводится по теоретическому материалу дисциплины. Максимальное количество баллов за собеседование – 10 баллов.

Понимание проблемы, стремление разъяснить ее суть с научно-методических позиций Грамотность речи и логичность изложения материала.	10 баллов
--	-----------

Критерии и шкала оценивания итогового теста

Итоговый тест включает:

1) Теоретические вопросы различного характера (открытые задания, задания с единственным выбором ответа, задания на установление соответствия и последовательности действий), позволяющие оценить знание программного материала дисциплины;

2) Задания, позволяющие оценить умение выполнять предусмотренные программой типовые задания. Максимальное количество баллов за итоговый тест – 12 баллов

2.1. Критерии и шкалы оценивания результатов обучения при проведении промежуточной аттестации

Промежуточная аттестация предназначена для определения уровня освоения всего объема учебной дисциплины. Для оценивания результатов обучения при проведении промежуточной аттестации используется четырехбалльная шкала

Основные виды систем оценивания

Европейская	100-балльная	4-балльная	2-балльная
A	94-100	отлично	зачтено
A-	90-94		
B+	85-89		
B	80-84	хорошо	
B-	75-79		
C+	70-74		
C	65-69	удовлетворительно	
C-	60-64		
D	55-59		
F	50-54	неудовлетворительно	не зачтено

Шкала оценивания	Критерии	Уровень освоения компетенций
Отлично	<i>наличие глубоких и исчерпывающих знаний в объеме пройденного программного материала, правильные и уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, знание дополнительно рекомендованной литературы</i>	Эталонный

<i>Хорошо</i>	<i>наличие твердых и достаточно полных знаний программного материала, незначительные ошибки при освещении заданных вопросов, правильные действия по применению знаний на практике, четкое изложение материала</i>	<i>Стандартный</i>
<i>Удовлетворительно</i>	<i>наличие твердых знаний пройденного материала, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов, правильные действия по применению знаний на практике</i>	<i>Пороговый</i>
<i>Неудовлетворительно</i>	<i>наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.</i>	<i>Компетенции не сформированы</i>

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1. Оценочные средства текущего контроля успеваемости

Контрольная работа № 1

Разработать программу, выполняющую операции над целочисленным динамическим массивом из n элементов ($n \leq 100$):

- ввод массива;
- вывод массива на экран;
- дополнительные операции.

Алгоритмы выполнения операций оформить в виде функций. В алгоритмах выполнения операций добавления и удаления элементов предусмотреть проверку возможности выполнения операций. Программа должна выводить текстовое меню для тестирования операций.

Варианты заданий приведены в табл. 1.

Таблица 1

Варианты заданий

№	Дополнительные операции	
	Операция 1	Операция 2
1	Сумма положительных чисел массива	Добавление после первого максимального числа числа x
2	Произведение отрицательных чисел массива	Удаление всех максимальных чисел
3	Количество четных чисел массива	Добавление перед первым максимальным числом числа x
4	Среднее арифметическое чисел массива	Удаление числа перед последним максимальным числом
5	Количество чисел массива, принадлежащих диапазону $[x, y]$	Добавление после последнего максимального числа числа x
6	Произведение чисел массива с нечетными индексами	Удаление числа после последнего максимального числа
7	Перестановка чисел массива в обратном порядке	Добавление перед последним максимальным числом числа x

8	Поиск номера последнего отрицательного элемента	Удаление числа перед каждым максимальным числом
9	Поиск максимального элемента	Добавление после первого отрицательного числа числа x
10	Замена отрицательных элементов массива на их модули	Добавление перед каждым максимальным числом числа x
11	Поиск номера первого максимального элемента	Добавление перед каждым нулевым числом числа x
12	Поиск номера последнего минимального элемента	Добавление после каждого отрицательного числа числа x
13	Количество чисел массива отличных от последнего элемента массива	Удаление числа перед первым максимальным числом
14	Сумма элементов массива с четными индексами	Добавление после каждого максимального числа числа x
15	Сумма четных чисел	Удаление числа после первого максимального числа

Расчетное задание по дисциплине «Программирование», 4 семестр, 12 баллов

Расчетное задание состоит из двух частей

I часть – динамические структуры данных.

II часть – объектно-ориентированное программирование.

I часть. Используя динамические структуры данных, разработать программу согласно варианту.

Предусмотреть:

- защиту от неправильного ввода данных;- ввод исходных данных с клавиатуры и из текстового файла.

Задание: Матрица размера $N \times M$ определяет некоторый лабиринт: элемент 1 обозначает стену, а 0 определяет свободное место. В первой строке матрицы определяются входы $x(i)$, а в последней выходы $y(i)$, $i=1, \dots, k$, которые должны быть нулевыми элементами. Необходимо определить, можно ли:

а) провести k человек от входа $x(i)$ до выхода $y(i)$ соответственно, $i=1, \dots, k$, таким образом, чтобы каждое свободное место посещалось не более одного раза.

б) то же, но человека можно выводить через любой из выходов. Примечание: Движение в лабиринте осуществляется только по вертикали или горизонтали. Примечание. Для существования решения необходимо проводить людей от самого левого входа к самому левому выходу, от самого правого - к правому и т.д. Поэтому рассмотрим только случай б) Основная стратегия человека, вошедшего в самый левый вход, состоит в прохождении лабиринта, используя наиболее левые свободные места лабиринта, т.е. он должен

двигаться, держась правой рукой за "стенку" лабиринта. Этот процесс можно формализовать следующим образом. Находясь в очередной позиции лабиринта, он должен помнить, с какой стороны он пришел сюда (слева, справа, сверху, снизу), и, руководствуясь этой информацией, выбрать следующее наиболее предпочтительное направление в новую позицию (куда он может пойти и где еще не был). При этом удобно использовать стек, в вершине которого находятся координаты текущей позиции и направление, по которому в нее пришли. При этом все посещенные клетки метятся. Тогда, если в позицию попали сверху, то наилучшим направлением будет налево, затем вниз, направо, и наконец назад (вверх). Аналогично можно определить наилучшие направления для других случаев Эта стратегия повторяется каждым из людей, при этом позиции, помеченные предыдущими людьми, считаются запрещенными для следующих людей.

II часть.

1. Разработать программное приложение для создания и работы с заданной фигурой (объектом).
2. Программа должна демонстрировать принципы ООП (полиморфизм, инкапсуляцию, наследование)
3. Программа должно выполнять следующие действия
 - изменение цвета объекта;
 - масштабирование объекта (увеличение, уменьшение);
 - движение объекта (вверх, вниз, влево, вправо, по диагонали, по выбранной траектории);-
 - поворот объекта.

Контрольный тест 4 семестра, дисциплина «Программирование»

1. Логическое выражение

$(N \text{ Mod } 10 \text{ Mod } 2) \text{ Or } (N \text{ Div } 10 \text{ Mod } 10 \text{ Mod } 2 = 0) \text{ Or } (N \text{ Div } 100 \text{ Mod } 2 = 0)$

должно принимать значение TRUE тогда и только тогда, когда истинно высказывание

- 1) в трёхзначном натуральном числе все цифры чётные; 2) в трёхзначном натуральном числе одна чётная цифра; 3) в трёхзначном натуральном числе две чётных цифры; 4) в трёхзначном натуральном числе хотя бы одна чётная цифра; 5) в трёхзначном натуральном числе нет чётных цифр.

2. Ошибку "Structure too large" (структура превышает максимально допустимый объём статической памяти) вызовет описание

- 1) `Type Vector = Array[Byte] Of Integer; Var C : Array[1..10] Of Vector;`
- 2) `Var T : File Of String;`
- 3) `Type A = Record S : String; A, B, C : Array[10..20] Of Real End; Var M : Array[1..5, 1..8] Of A;`
- 4) `Var K : Array [Byte, Byte] Of String[6]; 5) Var S : Array[-10000..10000] Of String[2].`

3. К процедурам для работы с динамическими переменными не относится

- 1) Mark; 2) New; 3) Release; 4) Seek; 5) Dispose.

4. Имеется описание

`Type A = Array[0..100] Of Real; B = ^A; Var M : Array[1..5] Of B;`

Для хранения массива M необходим объём памяти (байт)

- 1) 606; 2) 4; 3) 20; 4) 12120; 5) 6.

5. Фрагмент программы

```
К := 0;
While Not Eof(F) Do
Begin ReadLn(F, S); I := 1; While I <=
    Length(S) Do
    Begin If S[I] In ['А'..'Я', 'а'..'я', 'р'..'я']
        Then Begin К := К + 1;
                Delete(S, I, 1); I := I - 1
            End;
        I := I + 1
    End
End
End;
```

выполняет следующее действие:

- 1) удаляет из текстового файла F все русские буквы; 2) определяет в текстовом файле количество символов, являющихся русскими буквами; 3) определяет в текстовом файле количество символов, не являющихся русскими буквами; 4) определяет в текстовом файле количество символов; 5) удаляет из текстового файла F все символы, не являющиеся русскими буквами.

6. В фрагменте программы (здесь Var F : File Of Integer; I, K, Vsp : Integer;)

```
Reset(F); К := FileSize(F) - 1; For I := 0 To
К Do
Begin Seek(F, I); Read(F, Vsp); Seek(F, FileSize(F));
Write(F, Vsp) End;
```

выполняется

- 1) сортировка файла; 2) изменение порядка следования элементов на обратный; 3) дописывание в конец исходного файла полную его копию с сохранением порядка следования элементов; 4) дописывание в конец исходного файла полную его копию с изменением порядка следования элементов на противоположный; 5) не выполняется никаких действий по изменению файла.

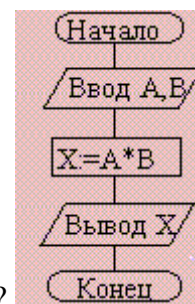
7. Имеется описание

Type Dn = (pn, vt, sr, cht, ptn, sb, vs); Mn = Set Of Dn; Var V : Mn;
и фрагмент программы

$$V := [pn..ptn] * [sr, ptn..vs] - [sb];$$

После исполнения этого фрагмента переменная V имеет значение

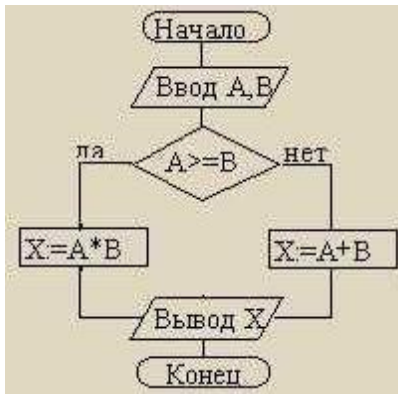
- 1) [pn..vs]; 2) [sr, ptn]; 3) [sb]; 4) []; 5) [pn..ptn].



8. Алгоритм какого типа изображен на блок-схеме?

- 1) циклический; 2) разветвляющийся; 3) вспомогательный; 4) линейный; 5) комбинация развилки и цикла.

9. После исполнения фрагмента программы, изображенного на блок-схеме,



при $A = 5$, $B = 4$ значение X будет равно

- 1) 20; 2) 9; 3) 5; 4) 4; 5) 1.

10. В приведенном фрагменте программы (N типа LongInt, $N > 0$)

```
P := 1;
While P <= N Do
Begin
    Left := N Div (P * 10) * (P * 10); Right := N Mod
    P;
    K := ((N Mod (P * 10) Div P + 1) Mod 10) * P;
    N := Left + K + Right; P := P * 10
End;
```

натуральное число N изменяется по следующему правилу

- 1) не изменяется; 2) в каждый разряд прибавляется 1; 3) из каждого разряда вычитается 1; 4) в каждый разряд прибавляется 1, если значение в разряде — не девять, иначе заменяется на нуль; 5) каждая девятка в десятичной записи числа заменяется на нуль.

11. Цикл с предусловием выполняется так:

- 1) выполняется тело цикла, изменяется параметр цикла, проверяется условие продолжения выполнения цикла; 2) изменяется параметр цикла, проверяется условие продолжения выполнения цикла, выполняется тело цикла; 3) проверяется условие продолжения выполнения цикла, выполняется тело цикла; 4) тело цикла выполняется N раз (N — натуральное); 5) определяется, сколько раз должен быть выполнен цикл, и далее цикл с предусловием сводится к циклу с параметром.

12. В текстовом файле каждая строка заканчивается

- 1) числами 10 и 13; 2) символами с кодами 10 и 13; 3) символом с кодом 13; 4) числом 0; 5) символом с кодом 10.

13. Процедуры ReadLn и WriteLn можно использовать при работе с

- 1) типизированными файлами; 2) нетипизированными файлами; 3) типизированными и нетипизированными файлами; 4) текстовыми файлами; 5) любыми файлами.

14. Значение выражения

$$\text{Ord}(x > y) + \text{Ord}(\text{Ord}(z = 'F'))$$

при $x = 7, y = 0, z = 'F'$ равно

- 1) TRUE; 2) FALSE; 3) 0; 4) 1; 5) 2.

15. Идентификатор в Turbo Pascal не может начинаться с

- 1) латинской буквы; 2) заглавной латинской буквы; 3) цифры; 4) знака подчёркивания; 5) латинской буквы, а затем знака подчёркивания.

16. В приведенном фрагменте программы (First — ссылка на первый элемент списка; список объявлен следующим образом: Type SS = ^List; List = Record A : LongInt; Next : SS End;)

```
P := First; S := 0; While Not (P = Nil) Do Begin S := S + 1; P := P^.Next End;
```

определяется

- 1) первый элемент списка; 2) сумма элементов списка; 3) сумма первого и последнего элементов списка; 4) количество элементов списка; 5) количество звеньев списка, где указатель на следующее звено не Nil.

17. При исполнении фрагмента программы

```
Var C : Integer;  
  Procedure R1(Var A : Integer; C : Boolean); Procedure R2;  
    Var C : String; Begin A  
      := 1 End;  
  Begin C := True; R2 End;  
Begin C := 100; R1(C, False); WriteLn(C) End.
```

будет напечатано значение переменной C

- 1) True; 2) 1; 3) 100; 4) неизвестно что, поскольку значение переменной C не определено; 5) False.

18. Цикл в фрагменте программы

```
P := 4; Repeat P := P * 0.1 Until P < 0.0001;
```

будет исполнен

- 1) 0 раз; 2) 1 раз; 3) 4 раза; 4) 5 раз; 5) бесконечное число раз.

19. Кодировается шестнадцатибитовое целое со знаком (тип Integer).
11111111111110000₂ — это код числа

- 1) -15; 2) 15; 3) 16; 4) -16; 5) -30000.

20. Свойством алгоритма является

- 1) результативность; 2) цикличность; 3) возможность изменения последовательности выполнения команд; 4) возможность выполнения алгоритма в обратном порядке; 5) простота при записи на языках программирования.

21. Из перечисленных ниже в программе обязателен

- 1) раздел Var; 2) раздел Const; 3) раздел Type; 4) раздел Label; 5) раздел Begin ... End.

22. Ввод данных — это

- 1) процесс передачи данных из оперативной памяти на внешний носитель;
- 2) процесс ввода с клавиатуры каких-либо значений;
- 3) передача данных от внешнего носителя в оперативную память для обработки;
- 4) присваивание конкретных значений переменным, которые используются в программе;
- 5) запись файла на диск.

23. Значение R после выполнения операции логического присваивания

$$R := \text{Not} (A \text{ Or } B \text{ Or } (X > 2) \text{ And } (Y < 0))$$

при $A = \text{False}$, $B = \text{False}$, $X = 3$, $Y = 2$ будет равно

- 1) -1;
- 2) False;
- 3) True;
- 4) 0;
- 5) 1.

24. С помощью какой из приведенных серий команд переменной B присваивается значение выражения?

$$\left(\frac{x+y}{x-y} - \frac{x-y}{x+y} \right) \cdot \left(\frac{x-y}{x+y} + \frac{x+y}{x-y} \right)$$

а) $A := (x + y) / (x - y)$; $B := (A - 1 / A) * (1 / A + A)$;

б) $A := (x + y) / (x - y)$; $B := \text{Sqr}(A) - \text{Sqr}(1 / A)$;

в) $A := (x - y) / (x + y)$; $B := \text{Sqr}(1 / A) - \text{Sqr}(A)$;

- 1) а;
- 2) б;
- 3) в;
- 4) всех трех;
- 5) ни один из ответов 1–4 не является верным.

25. Значения переменных a и b после выполнения следующих действий

$$a := 15 \text{ Div } (16 \text{ Mod } 7); b := 34 \text{ Mod } a * 5 - 29 \text{ Mod } 5 * 2;$$

будут равны

- 1) $a = 1$, $b = 160$;
- 2) $a = 1$, $b = 4$;
- 3) $a = 7$, $b = 25$;
- 4) $a = 7$, $b = 22$;
- 5) $a = 7$, $b = 28$.

26. Во фрагмент алгоритма

```
For K := 10 To 99 Do
Begin
  P1 := K Div 10; P2 :=
  K Mod 10; S := P1 +
  P2;
  If _____ Then WriteLn(K)
End;
```

печатающий все двузначные числа, в записи которых есть цифра N или сумма цифр которых равна самим числам, нужно вписать логическое выражение

- 1) $(P1 = N) \text{ Or } (P2 = N) \text{ And } (S = K)$;
- 2) $(P1 = N) \text{ Or } (P2 = N) \text{ Or } (S = K)$;
- 3) $(P1 = N) \text{ And } (P2 = N) \text{ Or } (S = K)$;
- 4) $((P1 = N) \text{ Or } (P2 = N)) \text{ And } (S = K)$;
- 5) $(P1 = N) \text{ And } (P2 = N) \text{ And } (S = N)$.

27. Значения переменных p и d после выполнения фрагмента алгоритма

```
k := 47; Case k Mod 9 Of
  5: Begin d := k; p := True End;
  0..2: Begin d := 2; p := False End;
  8: Begin d := 1; p := False End
Else
  Begin d := 1; p := True End
End;
```

равны

1) $p = \text{True}, d = 1$; 2) $p = \text{False}, d = 2$; 3) $p = \text{False}, d = 3$; 4) $p = \text{True}, d = 47$; 5) $p = \text{True}, d = 2$.

28. Тело цикла в программе

```
a := 1; b := 1; While a + b < 8 Do begin a := a + 1; b := b + 2 end;
```

выполнится

1) 1 раз; 2) 2 раза; 3) 3 раза; 4) ни разу; 5) бесконечное число раз.

29. Элементы массива $p[1..5]$ равны соответственно 1, -1, 5, 2, 4. Значение выражения

$$p[1] * p[3] - p[2] * p[2] + p[p[5] - p[2]]$$

равно

1) 8; 2) -8; 3) 12; 4) -12; 5) 6.

30. Задана строка St . Фрагмент алгоритма

```
S := 0;
For I := 1 To Length (St) Do Begin
    Val(St[I], d, k);
    If K = 0 Then S := S + d
End;
```

1) определяет количество цифр в строке; 2) подсчитывает количество нулей в строке; 3) определяет сумму номеров позиций в строке, где стоят цифры; 4) подсчитывает сумму цифр в строке; 5) определяет сумму номеров позиций в строке, где стоят нули.

31. Какая из приведенных серий операторов определяет и печатает индекс последнего отрицательного элемента в линейном массиве из n элементов?

а) $i := n$; While $(i >= 1) \text{ And } (m[i] > 0) \text{ Do Dec } (i)$; If $i < 1$ Then WriteLn ('i = 0') Else WriteLn ('i = ', i);
б) $k := 0$; For $i := 1$ To n Do If $m[i] < 0$ Then $k := i$; WriteLn ('i = ', k); в) $i := n$; Repeat $i := i - 1$ Until $(m[i] < 0)$; WriteLn ('i = ', i);

1) а, б; 2) б, в; 3) а, б, в; 4) б; 5) ни один из ответов 1–4 не верен.

32. Задан линейный массив $M[1..n]$.

```
Function Control (M: Myarray): Boolean; Var I : Integer;
Begin I := 1;
    While (I <= n) And (M[I] > 0) Do Inc(I); Control := (I <= n);
End;
```

Если в данном массиве все элементы положительные, приведенная функция возвращает значение

1) n ; 2) True; 3) False; 4) $I <= n$; 5) ни один из ответов 1–4 не верен.

33. Задан двумерный массив $X[1..n, 1..m]$. Процедура

```
Procedure Sub (Var X: Myarray); Var i, j: Integer;
```

```

Begin For i := 1 To n Do
    For j := 1 To m Div 2 Do X[i, 2 * j] := X[i, 2 * j] + X[i, 1];
End;

```

1) к элементам столбцов в первой половине матрицы прибавляет элементы первого столбца соответствующей строки; 2) добавляет к матрице еще M столбцов с элементами, равными соответствующим элементам первого столбца; 3) к элементам четных столбцов прибавляет элементы первого столбца соответствующей строки; 4) к элементам четных строк прибавляет элементы первой строки соответствующего столбца; 5) меняет порядок столбцов таблицы.

34. Задан двумерный массив $X[1..n, 1..m]$. Функция

```

Function Check (X: Myarray): Boolean; Var i, j :
Integer; t : Boolean; Begin t := True; i := 1;
    While t And (i <= n) Do
        Begin    j:= 1; While (j <= m) And (X[i, j] <> 0) Do
Inc (j);
                                t:= (j = m + 1); Inc (i)
        End;
    Check := Not t
End;

```

Inc (j);

End;

возвращает значение

1) True, если все элементы массива ненулевые; 2) True, если в массиве есть элемент, равный нулю; 3) False, если в массиве есть элемент, равный нулю; 4) Not t; 5) ни один из ответов 1–4 не верен.

35. Среди перечисленных соответствий, которые необходимо соблюдать между формальными и фактическими параметрами

а) соответствие по типу параметров; б) соответствие по количеству параметров; в) соответствие по типу используемых вспомогательных переменных; г) соответствие по порядку перечисления;

лишним является

1) а; 2) б; 3) в; 4) г; 5) ни один из ответов 1–4 не верен.

36. Определите тип выражения (здесь $A : \text{Array}[1..20] \text{ Of Real}$; $B : \text{Boolean}$; $C : \text{Integer}$)

```

C + Ord(Round(A[7]) + Ord(B)) – Trunc(A[1])

```

1) Real; 2) Integer; 3) Boolean; 4) Byte; 5) среди ответов 1–4 нет верного.

37. Список объявлен следующим образом

```

Type Ukaz = ^Zveno; Zveno = Record X : String; N :

```

Ukaz End;

```

    Var First : Ukaz; {ссылка на начало списка}

```

В следующем фрагменте программы

```

P:= First;
While P^.N <> Nil Do Begin
    B:= P; M := P;

```

While B <> Nil Do Begin If B^.X < M^.X Then M:= B; B:=
 B^.N End;
 S:= P^.X; P^.X:= M^.X; M^.X:= S; P:= P^.N
 End;
 выполняется

- 1) перемещение компонента к началу списка; 2) сортировка компонентов списка в порядке возрастания; 3) сортировка компонентов списка в порядке убывания; 4) перестановка соседних компонентов списка; 5) добавление в список нескольких новых компонент.

38. Фрагмент программы

S := A; A := B; B := S;

выполняет

- 1) обмен значений переменных A, B; 2) присваивание переменным A, B значения S; 3) замена значения переменной A значением переменной B; 4) во фрагменте не выполняется никаких действий; 5) замена значения переменной B значением переменной A.

39. Имеется следующее описание

Type U = ^Zveno; Zveno = Record X, Y : Boolean;
 Pred, Next : U End;
 Var Logic : Boolean; A, B : Pointer; X, Y : U;

К ошибке компиляции "Несовместимость типов" приведет следующее присваивание

- 1) A := X^.Next^.Next; 2) X := Y; 3) Logic := X^.X; 4) X^.Next := A; 5) X^ := Y^.Next.

40. При присваивании изменяется

- 1) алгоритм; 2) имя переменной; 3) тип переменной; 4) значение переменной; 5) значение константы.

Инструкция по оценке заданий тест-билета (ключи ответов)

№ задания	1	2	3	4	5	6	7	8	9	10
№ ответа	4	4	4	3	2	3	2	4	1	4
№ задания	11	12	13	14	15	16	17	18	19	20
№	3	2	4	5	3	4	2	4	4	1

ответа										
№ задания	21	22	23	24	25	26	27	28	29	30
№ ответа	5	3	3	4	4	2	2	2	5	4
№ задания	31	32	33	34	35	36	37	38	38	40
№ ответа	1	3	3	2	3	2	2	1	5	4

Примерные темы расчетно-графических работ, 5 семестр, итоговый балл 10 баллов

1. Линейное программирование: решение задач графическим способом
2. Нарисовать графики функции вблизи начала координат при нескольких различных значениях параметров, выделив их различными цветами или стилями линий.

Индивидуальные задания №1, 4 семестр

Вариант 1.

1. Составить программу формирования прямоугольного массива А из n строк и m столбцов случайными числами, сделать возможным его редактирование. Вычислить сумму элементов k-го столбца.
2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве StringGrid.Cells - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве StringGrid.Cells - буквой "н". Вычислить средний балл для каждого учащегося и средний балл всей группы.

Вариант 2.

1. Составить программу формирования прямоугольного массива А из n строк и m столбцов случайными числами, сделать возможным его редактирование. Вычислить сумму элементов k-й строки.
2. Дана сводная ведомость выплаты заработной платы n рабочим за m месяцев (номер строки соответствует порядковому номеру рабочего, номер столбца - порядковому номеру месяца). Определить номер рабочего с максимальным доходом за весь отчетный период.

Вариант 3.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Вычислить среднее арифметическое элементов k -го столбца.
2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` - буквой "н". Вычислить средний балл за каждый урок и средний балл группы за весь период.

Вариант 4.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Определить количество положительных, отрицательных и нулевых элементов в массиве.
2. Дана сводная ведомость выплаты заработной платы n рабочим за m месяцев (номер строки соответствует порядковому номеру рабочего, номер столбца - порядковому номеру месяца). Определить номер рабочего с минимальным доходом за весь отчетный период.

Вариант 5.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Определить номер строки с максимальной суммой элементов.
2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` - буквой "н". Вычислить сумму пропущенных часов для каждого учащегося, общую сумму по всей группе и среднее значение (час/уч-ся). Учесть, что продолжительность каждого занятия составляет 2 часа.

Вариант 6.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Определить номер столбца с минимальной суммой элементов.
2. Даны необходимые данные для создания ведомости начисления заработной платы для n служащих некоторой организации. Номер строки соответствует порядковому номеру служащего в алфавитном списке, 1-й столбец содержит квалификационный разряд ($1 \leq r \leq 16$), 2-й столбец - коэффициент ставки ($0,5 \leq k \leq 2$), 3-й столбец - процент

премиальных ($0 \leq p \leq 50$). Рассчитать сумму оклада для каждого служащего $S_o = S_b * r * k$; сумму премиальных $S_p = S * p / 100$ и общую сумму начисления $S = S_o + S_p$. S_b - это текущий размер базовой величины (задается как исходное данное).

Вариант 7.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Определить номер строки с наименьшим количеством положительных элементов.
2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` буквой "н". Вычислить сумму пропущенных часов на каждом уроке, общую сумму за весь период и среднее значение (час/уч-ся). Учесть, что продолжительность каждого занятия составляет 2 часа.

Вариант 8.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Определить номер столбца с наибольшим количеством отрицательных элементов.
2. Даны необходимые данные для создания ведомости отчислений из заработной платы для n служащих некоторой организации. Номер строки соответствует порядковому номеру служащего в алфавитном списке, 1-й столбец содержит значение начисленной суммы S . Рассчитать для каждого служащего прогрессивный процент подоходного налога P_n (9% - если $S \leq 500$ тыс. руб.; 12% - если 500 тыс. руб. $< S \leq 1.5$ млн. руб.; 15% - если $S > 1.5$ млн. руб.), сумму подоходного налога ($S_n = S * P_n / 100$), сумму отчислений в пенсионный фонд S_p (1% от S) и сумму к выдаче ($S_v = S - S_n - S_p$).

Вариант 9.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Поменять местами i -ую и j -ую строку.
2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` - буквой "н". Определить количество уч-ся, имеющих хотя бы одну неудовлетворительную оценку (1 - 4) и их процент от общего числа уч-ся.

Вариант 10.

1. Составить программу формирования прямоугольного массива A из n строк и m столбцов случайными числами, сделать возможным его редактирование. Поменять местами i -ый и j -ый столбец.

2. Дана таблица результатов футбольного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i -й командой в игре с j -й командой (0 - проигрыш; 1 - ничья; 3 - победа). По главной диагонали в строковой таблице `StringGrid` записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Если матча еще не было, то соответствующие ячейки в строковой таблице `StringGrid` остаются пустыми, а в численном массиве в эти ячейки можно записать -2. Определить сумму набранных очков каждой командой и определить текущего лидера (или несколько лидеров с одинаковой суммой очков). Перед обработкой необходимо проверить корректность данных в таблице.

Вариант 11.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить сумму элементов, стоящих на главной и побочной диагонали.

2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` - буквой "н". Определить количество уч-ся, у которых нет оценок ниже 7 и их процент от общего числа уч-ся.

Вариант 12.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить сумму элементов, стоящих выше главной и побочной диагонали.

2. Дана таблица результатов футбольного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i -й командой в игре с j -й командой (0 - проигрыш; 1 - ничья; 3 - победа). По главной диагонали в строковой таблице `StringGrid` записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Если матча еще не было, то соответствующие ячейки в строковой таблице `StringGrid` остаются пустыми, а в численном массиве в эти ячейки можно записать -2. Определить для каждой команды количество побед, ничьих и поражений. Перед обработкой необходимо проверить корректность данных в таблице.

Вариант 13.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить сумму элементов, стоящих ниже главной и выше побочной диагонали.

2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве StringGrid.Cells - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве StringGrid.Cells - буквой "н". Определить количество уч-ся, имеющих только отличные оценки (9 - 10) и их процент от общего числа уч-ся.

Вариант 14.

1. Составить программу формирования квадратной матрицы А n-го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить сумму элементов, стоящих выше главной и ниже побочной диагонали.

2. Дана итоговая таблица результатов футбольного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i-й командой в игре с j-й командой (0 - проигрыш; 1 - ничья; 3 - победа). По главной диагонали в строковой таблице StringGrid записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Определить 2 команды с худшим результатом, которые покидают турнир. Обеспечить корректность данных в таблице.

Вариант 15.

1. Составить программу формирования квадратной матрицы А n-го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить количество четных элементов в заданной строке.

2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве StringGrid.Cells - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве StringGrid.Cells - буквой "н". Упорядочить данные в порядке убывания среднего балла уч-ся, сохранив при этом их исходные порядковые номера.

Вариант 16.

1. Составить программу формирования квадратной матрицы А n-го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить количество нечетных элементов в заданном столбце.

2. Дана итоговая таблица результатов футбольного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i-й командой в игре с j-й командой (0 - проигрыш; 1 - ничья; 3 - победа). По главной диагонали в строковой таблице StringGrid записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Определить 2 команды с лучшим результатом, которые выходят из турнира в плей-офф. Перед обработкой необходимо проверить корректность данных в таблице.

Вариант 17.

1. Составить программу формирования квадратной матрицы А n-го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить

количество четных элементов выше k -ой строки и выше главной диагонали.

2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` - буквой "н". Упорядочить данные в порядке возрастания количества пропущенных часов уч-ся, сохранив при этом их исходные порядковые номера. Учесть, что продолжительность одного занятия составляет 2 часа.

Вариант 18.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Вычислить количество нечетных элементов ниже k -ой строки и выше главной диагонали. Дана итоговая таблица результатов футбольного турнира среди N команд в два круга (номер строки соответствует номеру команды, в столбцах последовательно записаны: количество побед, ничьих, поражений, забитых и пропущенных голов). Вычислить для каждой команды сумму набранных очков (0 - проигрыш; 1 - ничья; 3 - победа), разность забитых и пропущенных голов, упорядочить данные в порядке занятых мест, сохранив исходные номера команд (при равенстве очков учитывается разность забитых и пропущенных голов). Перед обработкой необходимо проверить корректность данных в таблице (количество игр у каждой команды должно быть равно $N-2$, суммарная разность забитых и пропущенных голов должна равняться 0).

Вариант 19.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Построить линейный массив B из модулей отрицательных элементов матрицы A .

2. Дана одна страница учебного журнала некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру урока). Отсутствие оценки за урок в численном массиве обозначить значением 0, в строковом массиве `StringGrid.Cells` - пустая строка. Отсутствие учащегося на уроке в численном массиве обозначить значением -1, в строковом массиве `StringGrid.Cells` - буквой "н". Определить количество оценок для каждого уч-ся и среднее значение количества оценок на одного уч-ся по всей группе.

Вариант 20.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Построить линейный массив B из средних арифметических строк матрицы A .

2. Дана таблица результатов хоккейного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i -й командой в игре с j -й командой (0 - поражение; 1 - поражение в овертайме или по булидам; 2 - победа в овертайме или по булидам; 3 - победа). По главной диагонали в строковой таблице `StringGrid` записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Если матча еще не было, то соответствующие ячейки в строковой таблице `StringGrid` остаются пустыми, а в численном массиве в эти ячейки можно записать -2. Определить сумму набранных очков

каждой командой и определить текущего лидера (или несколько лидеров с одинаковой суммой очков). Перед обработкой необходимо проверить корректность данных в таблице.

Вариант 21.

1. Составить программу формирования квадратной матрицы A n -го порядка (n строк и n столбцов) случайными числами, сделать возможным ее редактирование. Построить линейный массив B из сумм положительных элементов в столбцах матрицы A .

2. Дана сводная ведомость оценок за семестр уч-ся некоторой группы (номер строки соответствует порядковому номеру учащегося в алфавитном списке, номер столбца - порядковому номеру предмета). Вычислить средний балл (S_b) для каждого уч-ся за семестр и коэффициент для начисления стипендии (k), который равен: 0, если $S_b < 5$; 1, если $5 \leq S_b \leq 6$; 1.2, если $6 < S_b \leq 7$; 1.4, если $7 < S_b \leq 9$; 1.8, если $S_b > 9$

Вариант 22.

1. Составить программу формирования квадратных матриц A и B n -го порядка (n строк и n столбцов) случайными числами, сделать возможным их редактирование. Вычислить сумму матриц (матрица C , в которой $C_{ij} = A_{ij} + B_{ij}$).

2. Дана таблица результатов хоккейного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i -й командой в игре с j -й командой (0 - поражение; 1 - поражение в овертайме или по булидам; 2 - победа в овертайме или по булидам; 3 - победа). По главной диагонали в строковой таблице StringGrid записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Если матча еще не было, то соответствующие ячейки в строковой таблице StringGrid остаются пустыми, а в численном массиве в эти ячейки можно записать -2. Определить для каждой команды количество чистых побед, побед и поражений в овертайме или по булидам и поражений. Перед обработкой необходимо проверить корректность данных в таблице.

Вариант 23.

1. Составить программу формирования смежных матриц A (m строк, n столбцов) и B (n строк, l столбцов) случайными числами, сделать возможным их редактирование. Вычислить произведение матриц, это матрица C , в которой:

$$C_{ij} = \sum_{k=1}^n A_{ik} \cdot B_{kj}, \text{ для } i=1..m, j=1..l$$

2. Дана сводная ведомость выплаты заработной платы n рабочим за m месяцев (номер строки соответствует порядковому номеру рабочего, номер столбца - порядковому номеру месяца). Вычислить размер средней заработной платы за месяц для каждого рабочего и общую по всему списку.

Вариант 24.

1. Составить программу формирования матриц A и B (n строк, m столбцов) случайными числами, сделать возможным их редактирование. Вычислить матрицу $C = 2A + 3B$.

2. Дана итоговая таблица результатов хоккейного турнира (номера строк и столбцов определяют номер команды). В ячейке (i, j) записано количество очков полученных i -й командой в игре с j -й командой (0 - поражение; 1 - поражение в овертайме или по булидам; 2 - победа в овертайме или по булидам; 3 - победа). По главной диагонали в строковой таблице StringGrid записывается символ "x" (команда не играет сама с собой), в численном массиве на главной диагонали можно записать -1. Определить 2 команды с лучшим результатом, которые проходят в плей-офф. Перед обработкой необходимо проверить корректность данных в таблице.

Вариант 25.

1. Составить программу формирования матрицы A (n строк, m столбцов) случайными

числами, сделать возможным ее редактирование. Получить вектор B , присвоив его k -му элементу значение 0, если все элементы k -го столбца матрицы A нулевые, иначе 1.

2. Дана сводная ведомость выплаты заработной платы n рабочим за m месяцев (номер строки соответствует порядковому номеру рабочего, номер столбца - порядковому номеру месяца). Вычислить размер средней заработной платы за каждый месяц и общую по всему списку.

Вариант 26.

1. Составить программу формирования матрицы A (n строк, m столбцов) случайными числами, сделать возможным ее редактирование. Получить вектор B , присвоив его k -му элементу значение 1, если элементы k -й строки матрицы A упорядочены по убыванию, иначе 0.

2. Дана итоговая таблица результатов футбольного турнира среди N команд в один круг (номер строки соответствует номеру команды, в столбцах последовательно записаны: количество побед, побед в овертайме или по булидам, поражений в овертайме или по булидам, поражений, забитых и пропущенных шайб). Вычислить для каждой команды сумму набранных очков (0 - поражение; 1 - поражение в овертайме или по булидам; 2 - победа в овертайме или по булидам; 3 - победа), разность забитых и пропущенных шайб, упорядочить данные в порядке занятых мест, сохранив исходные номера команд (при равенстве очков учитывается разность забитых и пропущенных шайб). Перед обработкой необходимо проверить корректность данных в таблице (количество игр у каждой команды должно быть равно $N-1$, суммарная разность забитых и пропущенных голов должна равняться 0).

Примерные темы групповых проектов. 5 семестр

- Разработка тестирующей программы
- Разработка чат бота
- Разработка собственного класса
- Создать класс «Объекты графики». Основными представителями класса являются объекты: прямая, прямоугольник, закрашенный прямоугольник или круг, замкнутая область. Основными операциями над объектами класса «Объекты графики» являются: наложение объектов, перемещение объектов, масштабирование.

3.2. Оценочные средства промежуточной аттестации

Вопросы к экзамену по дисциплине (4 семестр).

1. Этапы решения задач с использованием ЭВМ.
2. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
3. Понятие алгоритма. Понятие исполнителя. Система команд исполнителя.
4. Понятие величины. Типы величин. Присваивание величин. Совместимость по присваиванию.
5. Понятие о структурном программировании. Другие парадигмы программирования: сравнительная характеристика.
6. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика).

Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).

7. Система программирования Pascal.
8. Структура программы, элементы языка (алфавит). Понятие типа данных.
9. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
10. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.
11. Условный оператор. Оператор множественного ветвления (выбора).
12. Циклы в Pascal: с предусловием, с постусловием. Связь с другими циклами.
13. Циклы в Pascal: с параметром. Связь с другими циклами.
14. Структурированные типы данных. Линейные массивы. Примеры задач.
15. Структурированные типы данных. Двумерные массивы. Примеры задач.
16. Сортировка массивов. Метод выбора. Двоичный поиск в массиве.
17. Сортировка массивов. Метод обмена.
18. Сортировка массивов. Метод вставок.
19. Подпрограммы в Pascal. Основные способы передачи параметров в подпрограмму, их сравнение.
20. Подпрограммы в Pascal. Область видимости. Локальные и глобальные идентификаторы.
21. Процедуры. Организация и вызов. Примеры.
22. Функции. Организация и вызов. Примеры.
23. Простые типы данных в Pascal.
24. Структурированные типы данных. Строковый тип данных в Pascal: основные процедуры и функции, примеры.
25. Рекурсия. Механизм рекурсии. Примеры.

Вопросы к экзамену по дисциплине (5 семестр)

1. Этапы решения задач с использованием ЭВМ.
2. Понятие алгоритма. Подходы к определению алгоритма. Свойства алгоритма. Способы записи алгоритма.
3. Понятие алгоритма. Понятие исполнителя. Система команд исполнителя.
4. Понятие величины. Типы величин. Присваивание величин. Совместимость по присваиванию.
5. Понятие о структурном программировании. Другие парадигмы программирования: сравнительная характеристика.
6. Языки программирования. Алгоритмические языки (алфавит, синтаксис, семантика). Способы описания синтаксиса (язык металингвистических формул, синтаксические диаграммы).
7. Система программирования Pascal.
8. Структура программы, элементы языка (алфавит). Понятие типа данных.
9. Операции (арифметические, логические) на типах. Стандартные функции. Выражения.
10. Процедуры консольного ввода и вывода, управление вводом-выводом. Оператор присваивания. Совместимость по присваиванию.
11. Условный оператор. Оператор множественного ветвления (выбора).
12. Циклы в Pascal: с предусловием, с постусловием. Связь с другими циклами.
13. Циклы в Pascal: с параметром. Связь с другими циклами.
14. Структурированные типы данных. Линейные массивы. Примеры задач.
15. Структурированные типы данных. Двумерные массивы. Примеры задач.
16. Сортировка массивов. Метод выбора. Двоичный поиск в массиве.
17. Сортировка массивов. Метод обмена.
18. Сортировка массивов. Метод вставок.
19. Подпрограммы в Pascal. Основные способы передачи параметров в подпрограмму, их сравнение.
20. Подпрограммы в Pascal. Область видимости. Локальные и глобальные идентификаторы.
21. Процедуры. Организация и вызов. Примеры.
22. Функции. Организация и вызов. Примеры. Простые типы данных в Pascal.
23. Структурированные типы данных. Строковый тип данных в Pascal: основные процедуры и функции, примеры.
24. Рекурсия. Механизм рекурсии. Примеры.
25. Сортировка массивов. Метод быстрой сортировки.
26. Множества в Pascal. Примеры.
27. Комбинированный тип данных (записи). Оператор присоединения. Записи с вариантами. Программирование типовых алгоритмов обработки записей.
28. Файловые типы в Pascal. Общие процедуры для работы с файлами. Компонентные (типизированные) файлы.
29. Текстовые файлы. Текст-ориентированные процедуры и функции. Типовые задачи.
30. Прямой и последовательный доступ к компонентам файла. Процедуры и функции, ориентированные на прямой доступ к компонентам файла.
31. Поиск в типизированных файлах. Сортировка файлов (на примере одного из методов).

32. Типизированные файлы. Файлы записей. Типовые алгоритмы обработки.
33. Статическая и динамически распределяемая память. Пример использования указателей.
34. Динамические структуры данных. Однонаправленный список. Процедуры обработки списка.
35. Динамические структуры данных. Двухнаправленный список. Процедуры обработки списка.
36. Динамические структуры данных. Кольцевой список (однонаправленный или двухнаправленный). Процедуры обработки списка.
37. Стек. Процедуры обработки.
38. Очередь. Процедуры обработки.
39. Двоичное дерево. Добавление в дерево и поиск в дереве.
40. Двоичное дерево. Удаление элемента из дерева.
41. Модуль. Общая структура модуля. Компиляция и подключение модуля.
42. Объектно-ориентированное программирование.
43. Пример реализации задачи на ООП в Pascal
44. Концепция и технологии .NET
45. Парадигма объектно-ориентированного программирования и ее предшественники
46. Терминология объектно-ориентированного программирования: класс, объект, пере
47. Три базовых понятия парадигмы объектно-ориентированного программирования
48. Инкапсуляция: абстракция, интерфейс и реализация
49. Инкапсуляция: средства защиты и доступа

4. Методические материалы, определяющие процедуру оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

4.1. Описание процедур проведения текущего контроля успеваемости студентов

В таблице представлено описание процедур проведения контрольно-оценочных мероприятий текущего контроля успеваемости студентов, в соответствии с рабочей программой дисциплины (модуля), и процедур оценивания результатов обучения с помощью спланированных оценочных средств.

Наименование оценочного средства	Описания процедуры проведения контрольно-оценочного мероприятия и процедуры оценивания результатов обучения
Компьютерное тестирование	Компьютерное тестирование проводится по результатам освоения разделов дисциплины во время практических занятий. Во время проведения тестирования пользоваться учебниками, справочниками, конспектами лекций, тетрадями для практических занятий не разрешено. Преподаватель на практическом занятии, предшествующем занятию проведения теста, доводит до обучающихся: темы, количество заданий в тесте время выполнения.
Проект	Группе выдается тема проекта по вариантам. С помощью проекта студент вместе с группой разрабатывает программу, тестируют и защищают её

4.2. Описание процедур проведения промежуточной аттестации

Экзамен

При определении уровня достижений обучающихся на экзамене обращается особое внимание на следующее:

- дан полный, развернутый ответ на поставленный вопрос;
- показана совокупность осознанных знаний об объекте, проявляющаяся в свободном оперировании понятиями, умении выделить существенные и несущественные признаки, причинно-следственные связи;
- знание об объекте демонстрируются на фоне понимания его в системе данной дисциплины (модуля) и междисциплинарных связей;
- ответ формулируется в терминах дисциплины (модуля), изложен литературным языком, логичен, доказателен, демонстрирует авторскую позицию обучающегося;
- теоретические постулаты подтверждаются примерами из практики.

Необходимое количество баллов за 4 семестр

Оценочное средство	Количество баллов
Выполнение лабораторных работ	32
Выполнение домашних работ	16
Итоговый тест	12
Контрольная работа	10
Расчетная работа	10
Итоговое собеседование	10
Индивидуальное задание по вариантам	10

Необходимое количество баллов за 5 семестр

Оценочное средство	Количество баллов
Выполнение лабораторных работ	40
Выполнение домашних работ	10
Контрольная работа	10
Расчетно графическая работа	10
Итоговое собеседование	10
Групповой проект	20