

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММИРОВАНИЕ

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль): «Электронный бизнес»

Форма обучения: очная, заочная

Квалификация (степень) выпускника: бакалавр

Срок обучения: очная форма – 4 г., заочная форма – 4 года 6 мес.

Вид учебной работы	Трудоемкость, часы (з.е.)	
	Очная форма	Заочная форма
1. Контактная работа обучающихся с преподавателем:	52(1,44)	14(0,39)
Аудиторные занятия, часов всего, в том числе:	50(1,38)	12(0,33)
• лекции	16(0,44)	2(0,06)
• лабораторные	34(0,94)	10(0,28)
Промежуточная аттестация (контактная работа)	2(0,06)	2(0,06)
2. Самостоятельная работа студентов, всего	40(1,11)	123(3,42)
• др. формы самостоятельной работы	40(1,11)	123(3,42)
3. Промежуточная аттестация: экзамен	52(1,44)	7(0,19)
Итого	144(4)	144(4)

Хрущева Л.Б. Программирование: Рабочая программа дисциплины (модуля). – Казань: Казанский кооперативный институт (филиал) Российского университета кооперации, 2019. – 47 с.

Рабочая программа по дисциплине (модулю) «Программирование» по направлению подготовки 38.03.05 Бизнес-информатика, направленность «Электронный бизнес» составлена Хрущевой Л.Б., ст. преподавателем кафедры естественных дисциплин сервиса и туризма Казанского кооперативного института (филиала) Российского университета кооперации в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по направлению подготовки «Бизнес-информатика», утвержденного приказом Министерства образования и науки Российской Федерации от «11» августа 2016 года, № 1002, и учебными планами по направлению подготовки 38.03.05 Бизнес-информатика, направленность (профиль) «Электронный бизнес» (год начала подготовки -2019).

Рабочая программа:

обсуждена и рекомендована к утверждению решением кафедры естественных дисциплин, сервиса и туризма Казанского кооперативного института (филиала) Российского университета кооперации от «13» марта 2019 г., протокол № 7.

одобрена Научно-методическим советом Казанского кооперативного института (филиала) от 03.04.2019, протокол № 5.

утверждена Ученым советом Российского университета кооперации от 18.04.2019, протокол № 4.

© АНОО ВО ЦС РФ
«Российский университет
кооперации» Казанский
кооперативный институт
(филиал), 2019
© Хрущева Л.Б., 2019

СОДЕРЖАНИЕ

1. Цели, задачи освоения дисциплины (модуля)	5
2. Место дисциплины (модуля) в структуре образовательной программы	5
3. Перечень планируемых результатов обучения по дисциплине (модулю).....	5
4. Объем дисциплины (модуля) и виды учебной работы	6
5. Содержание дисциплины (модуля).....	6
5.1. Содержание разделов, тем дисциплины (модуля).....	6
5.2. Междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами (модулями)	8
5.3. Разделы, темы дисциплины (модуля) и виды занятий.....	8
6. Лабораторный практикум	9
7. Практические занятия (семинары).....	10
8. Примерная тематика курсовых проектов (работ).....	10
9. Самостоятельная работа студента	10
10. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю).....	11
11. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля).....	12
12. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» », необходимых для освоения дисциплины (модуля)	12
13. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационно-справочных систем (при необходимости)	12
14. Описание материально–технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).....	13
15. Методические указания для обучающихся по освоению дисциплины.....	13
16. Методические рекомендации по организации изучения дисциплины для преподавателей, образовательные технологии.....	14
ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)	16
1. Паспорт фонда оценочных средств	17
1.1 Компетенции, формируемые в процессе изучения дисциплины:.....	17
1.2. Сведения об иных дисциплинах (преподаваемых, в том числе, на других кафедрах) участвующих в формировании данных компетенций:	17
1.3. Этапы формирования и программа оценивания контролируемой компетенции:.....	17
1.4. Показатели и критерии оценивания компетенций, шкала оценивания	20
2. Типовые контрольные задания для оценки результатов обучения по дисциплине и иные материалы для подготовки к промежуточной аттестации	22
2.1 Материалы для подготовки к промежуточной аттестации	22
2.2. Комплект экзаменационных билетов для проведения промежуточной аттестации.....	28
2.3. Критерии оценки для проведения экзамена по дисциплине	31
2.4. Методические материалы, определяющие процедуру оценивания по дисциплине.....	31
ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ И ТЕКУЩЕЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)	32
1. Материалы для текущего контроля	33
Комплект практических задач	33
Вариант № 1	33
2. Материалы для проведения текущей аттестации.....	40
Текущая аттестация 1	40
Комплект заданий для проведения текущей аттестации №1	40
(в форме самостоятельной работы)	40
Текущая аттестация 2	44

Комплект заданий для проведения текущей аттестации №2	44
(в форме контрольной работы).....	44

1. Цели, задачи освоения дисциплины (модуля)

Цель - формирование у будущих специалистов практических навыков по основам алгоритмизации вычислительных процессов и программированию решения экономических, вычислительных и других задач, развитие умения работы с персональным компьютером на высоком пользовательском уровне, обучение работе с научно-технической литературой и технической документацией по программному обеспечению ПЭВМ.

Задача - реализация требований, установленных в квалификационной характеристике в области анализа, создания, внедрения, сопровождения и применения средств математического обеспечения информационных систем предметной области.

2. Место дисциплины (модуля) в структуре образовательной программы

Дисциплина относится к базовой части дисциплин блока Б1 «Дисциплины (модули)»

Для изучения дисциплины необходимы знания, умения и владения навыками формируемые предшествующей дисциплиной «Информатика» на базе полного среднего общего образования.

3. Перечень планируемых результатов обучения по дисциплине (модулю)

Изучение дисциплины направлено на формирование у обучающихся следующих компетенций:

ОПК-3 - способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях;

ПК-18 - способность использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования.

В результате изучения дисциплины обучающийся должен:

Формируемые компетенции (код компетенции)	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций	Наименование оценочного средства
ОПК-3 ПК-18	Знать основные технологии программирования	Практические задачи
	Уметь проектировать, внедрять и организации эксплуатацию ИС и ИКТ	Самостоятельная работа
	Уметь разрабатывать алгоритмы решения.	
	Уметь программировать задачи обработки данных в	

Формируемые компетенции (код компетенции)	Планируемые результаты обучения по дисциплине (модулю), характеризующие этапы формирования компетенций	Наименование оценочного средства
	предметной области,	
	Уметь выполнять тестирование и отладку программ	
	Владеть методами и инструментальными средствами разработки программ.	Контрольная работа
	Владеть методами оформления программной документации	

4. Объем дисциплины (модуля) и виды учебной работы

очная форма обучения

Вид учебной деятельности	Часов	
	Всего	По семестрам
		2
1. Контактная работа обучающихся с преподавателем:	52	52
Аудиторные занятия всего, в том числе:	64	64
Лекции	16	16
Лабораторные работы	34	34
Промежуточная аттестация (контактная работа)	2	2
2. Самостоятельная работа студента всего, в том числе:	40	40
Другие виды самостоятельной работы	40	40
Вид промежуточной аттестации – экзамен	52	52
ИТОГО:	часов	144
Общая трудоемкость	зач. ед.	4

заочная форма обучения

Вид учебной деятельности	Часов	
	Всего	По курсам
		2
1. Контактная работа обучающихся с преподавателем:	14	14
Аудиторные занятия всего, в том числе:	12	12
Лекции	2	2
Лабораторные работы	10	10
Промежуточная аттестация (контактная работа)	2	2
2. Самостоятельная работа студента всего, в том числе:	123	123
Другие виды самостоятельной работы	123	123
Вид промежуточной аттестации – экзамен	7	7
ИТОГО:	часов	144
Общая трудоемкость	зач. ед.	4

5. Содержание дисциплины (модуля)

5.1. Содержание разделов, тем дисциплины (модуля)

Тема 1. Основы алгоритмизации.

1. Понятие алгоритма,
2. способы записи алгоритмов,
3. типы вычислительных процессов,

4. этапы подготовки и решения задач на ЭВМ

Тема 2. Введение в программирование на процедурном языке.

1. Скалярные типы данных языка Паскаль,
2. структура программы,
3. средства ввода-вывода,
4. управляющие операторы

Тема 3. Описание и обработка данных структурированного типа.

1. Объявление массивов.
2. Работа со строковыми типами данных.
3. Определение типа множества.
4. Свойства множеств.
5. Операции над множествами.
6. Объявление записей.
7. Обращение к элементам записи.
8. Запись с вариантами

Тема 4. Модульное программирование.

1. Структура программы.
2. Описание процедуры.
3. Оператор вызова процедуры.
4. Описание функции.
5. Указатель функции.
6. Формальные и фактические параметры.
7. Область действия имен.
8. Процедуры без параметров.
9. Предварительно определенные процедуры.
10. Рекурсивные процедуры и функции.
11. Понятие модуля.
12. Описание модуля.
13. Интерфейсная, исполняемая и инициализирующая части.

Тема 5. Динамические переменные и указатели.

1. Общие сведения о динамическом распределении памяти и динамических переменных.
2. Указатели, объявление ссылочных переменных.
3. Управление выделением и освобождением динамической памяти на логическом уровне.
4. Управление динамической памятью на физическом уровне.
5. Работа со стеками и очередями.
6. Создание и обработка однонаправленных и двунаправленных списков.

Тема 6. Работа с файлами.

1. Общие сведения о файлах.
2. Описание файлов.
3. Стандартные процедуры и функции для работы с файлами.
4. Текстовые файлы.
5. Файлы с типом.
6. Организация последовательного и прямого доступа.

5.2. Междисциплинарные связи с обеспечиваемыми (последующими) дисциплинами (модулями)

Дисциплина «Программирование» формирует ОПК-3, ПК-18 компетенции, необходимые в дальнейшем для формирования ПК-6 компетенции.

5.3. Разделы, темы дисциплины (модуля) и виды занятий

очная форма обучения

№ п/п	Наименование темы дисциплины (модуля)	Виды занятий, включая самостоятельную работу студентов (в часах)			
		Лекции	Лабораторные занятия	Самостоятельная работа	Всего
1.	Основы алгоритмизации	2	4	6	12
2.	Введение в программирование на процедурном языке	4	6	6	16
3.	Описание и обработка данных структурированного типа	4	6	6	16
4.	Модульное программирование	2	6	6	14
5.	Динамические переменные и указатели	2	6	8	16
6.	Работа с файлами	2	6	8	16
	Итого	16	34	40	90

заочная форма обучения

п/п	Наименование темы дисциплины (модуля)	Виды занятий, включая самостоятельную работу студентов (в часах)			
		Лекции	Лабораторные занятия	Самостоятельная работа	Всего
1.	Основы алгоритмизации	-	1	20	20
2.	Введение в программирование на процедурном языке	0,5	1	20	22,5
3.	Описание и обработка данных структурированного типа	0,5	6	20	26,5
4.	Модульное программирование	1	2	20	23
5.	Динамические переменные и указатели	-	-	23	23
6.	Работа с файлами	-	-	20	20
	Итого	2	10	123	135

6. Лабораторный практикум

Лабораторные занятия проводятся с целью формирования компетенций обучающихся, закрепления полученных теоретических знаний на лекциях и в процессе самостоятельного изучения обучающимися специальной литературы.

очная форма обучения

№ п/п	Наименование раздела, темы дисциплины (модуля)	Тематика лабораторных занятий	Трудоёмкость (час.)
1	Основы алгоритмизации	Разработка алгоритмов линейной, ветвящейся и циклической структуры.	4
2	Введение в программирование на процедурном языке	Введение в программирование. Примеры простых программ	2
3		Работа с числовыми, символьными, логическими и перечислимыми типами данных. Целочисленная арифметика.	4
4	Описание и обработка данных структурированного типа	Массивы: нахождение элементов с максимальным и минимальным значением, вычисление суммы, произведения элементов, удовлетворяющих заданному условию. Поиск в массиве, сортировка массивов.	2
5		Обработка строк: редактирование строки, нахождение подстроки, удовлетворяющей заданному условию.	2
6		Обработка массива записей	2
7	Модульное программирование	Процедуры и функции. Рекурсивные Подпрограммы	2
8		Создание личного модуля	4
9	Динамические переменные и указатели	Создание и обработка однонаправленных и двунаправленных связанных списков. Очереди, стеки.	6
10	Работа с файлами	Создание, просмотр, редактирование файла записей.	2
11		Работа с текстовыми файлами.	4
	Итого		34

заочная форма обучения

№ п/п	Наименование раздела, темы дисциплины (модуля)	Тематика лабораторных занятий	Трудоёмкость (час.)
1	Основы алгоритмизации	Разработка алгоритмов линейной, ветвящейся и циклической структуры.	1
2	Введение в программирование на процедурном языке	Введение в программирование. Примеры простых программ	1
		Работа с числовыми, символьными, логическими и перечислимыми типами данных. Целочисленная арифметика.	
3	Описание и обработка данных структурированного типа	Массивы: нахождение элементов с максимальным и минимальным значением, вычисление суммы, произведения элементов,	2

№ п/п	Наименование раздела, темы дисциплины (модуля)	Тематика лабораторных занятий	Трудоёмкость (час.)
		удовлетворяющих заданному условию. Поиск в массиве, сортировка массивов.	
		Обработка строк: редактирование строки, нахождение подстроки, удовлетворяющей заданному условию.	4
4	Модульное программирование	Процедуры и функции. Рекурсивные Подпрограммы	2
	Итого		6

7. Практические занятия (семинары)

Практические занятия не предусмотрены учебными планами.

8. Примерная тематика курсовых проектов (работ)

Курсовые проекты (работы) не предусмотрены учебными планами.

9. Самостоятельная работа студента

Тема 1. Основы алгоритмизации

Чтение текста (учебника, дополнительной литературы); работа с ресурсами Интернет. Конспектирование текста с составлением схем и таблиц. Работа с конспектом лекций. Ответы на контрольные вопросы.

Оценочное средство Опросы

Тема 2. Введение в программирование на процедурном языке

Чтение текста (учебника, дополнительной литературы); работа с ресурсами Интернет. Конспектирование текста с составлением схем и таблиц. Работа с конспектом лекций. Ответы на контрольные вопросы.

Оценочное средство Тесты

Тема 3. Описание и обработка данных структурированного типа

Чтение текста (учебника, дополнительной литературы); работа с ресурсами Интернет. Конспектирование текста с составлением схем и таблиц. Работа с конспектом лекций. Ответы на контрольные вопросы.

Оценочное средство Контрольные работы

Тема 4 Модульное программирование

Чтение текста (учебника, дополнительной литературы); работа с ресурсами Интернет. Конспектирование текста с составлением схем и таблиц. Подготовка к лабораторным работам. Подготовка отчетных документов по итогам выполнения лабораторных заданий

Оценочное средство Контрольные работы

Тема .5 Динамические переменные и указатели.

Чтение текста (учебника, дополнительной литературы); работа с ресурсами Интернет. Работа с конспектом лекций. Ответы на контрольные вопросы.

Оценочное средство Контрольные работы

Тема 6 Работа с файлами

Чтение текста (учебника, дополнительной литературы); работа с ресурсами Интернет. Работа с конспектом лекций. Ответы на контрольные вопросы. Подготовка к лабораторным работам. Подготовка отчетных документов по итогам выполнения лабораторных заданий

Оценочное средство Контрольные работы

10. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа студентов предполагает тщательное освоение учебной и научной литературы по изучаемой дисциплине.

При изучении основной рекомендуемой литературы студентам необходимо обратить внимание на выделение основных понятий, их определения, научно-технические основы, узловые положения, представленные в изучаемом тексте.

При самостоятельной работе студентов с дополнительной литературой необходимо выделить аспект изучаемой темы (что в данном материале относится непосредственно к изучаемой теме и основным вопросам).

Дополнительную литературу целесообразно прорабатывать после основной, которая формирует базис для последующего более глубокого изучения темы. Дополнительную литературу следует изучать комплексно, рассматривая разные стороны изучаемого вопроса. Обязательным элементом самостоятельной работы студентов с литературой является ведение необходимых записей: конспекта, выписки, тезисов, планов.

Для самостоятельной работы по дисциплине используется следующее учебно-методическое обеспечение:

а) основная литература

1. Математическое программирование: Учебник / Балдин К.В., Брызгалов Н.А., Рукосуев А.В., - 2-е изд. - М.: Дашков и К, 2018. - 218 с. - Режим доступа: <http://znanium.com/catalog/product/415097>

2. Программирование в среде Lazarus для школьников и студентов: Учебное пособие / Гуриков С.Р. - М.: Форум, НИЦ ИНФРА-М, 2016. - 336 с. - Режим доступа: <http://znanium.com/catalog/product/520628>

б) дополнительная литература

1. Программирование на языке Си: Учебное пособие /А.В. Кузин, Е.В. Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с. - Режим доступа: <http://znanium.com/catalog/product/505194>

11. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

а) нормативные документы

1. Об информации, информационных технологиях и защите информации: Федер. закон от 27.07.2006 N 149-ФЗ (ред. от 24.11.2014) . - КонсультантПлюс. [Электрон. ресурс]. – Электрон. дан. – [М., 2014].

2. Об электронной цифровой подписи: Федер. закон от 06.04.2011 N 63-ФЗ (ред. от 28.06.2014). - КонсультантПлюс. [Электрон. ресурс]. – Электрон. дан. – [М., 2014].

3. О персональных данных: Федер. закон от 27.07.2006 N 152-ФЗ (ред. от 21.07.2014) . - КонсультантПлюс. [Электрон. ресурс]. – Электрон. дан. – [М., 2014].

б) основная литература

1. Математическое программирование: Учебник / Балдин К.В., Брызгалов Н.А., Рукоусев А.В., - 2-е изд. - М.: Дашков и К, 2018. - 218 с. - Режим доступа: <http://znanium.com/catalog/product/415097>

2. Программирование в среде Lazarus для школьников и студентов: Учебное пособие / Гуриков С.Р. - М.: Форум, НИЦ ИНФРА-М, 2016. - 336 с. - Режим доступа: <http://znanium.com/catalog/product/520628>

в) дополнительная литература

1. Программирование на языке Си: Учебное пособие /А.В. Кузин, Е.В. Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с. - Режим доступа: <http://znanium.com/catalog/product/505194>

12. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» », необходимых для освоения дисциплины (модуля)

- <https://www.book.ru/> - ЭБС Book.ru
- <http://www.iprbookshop.ru> - ЭБС IPRbooks
- <https://ibooks.ru/> -ЭБС Айбукс.ru/ibooks.ru
- <https://rucont.ru/> - ЭБС «Национальный цифровой ресурс «Руконт»
- <http://znanium.com/> - ЭБС Znanium.com
- <https://dlib.eastview.com/>- База данных East View

13. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационно-справочных систем (при необходимости)

1. Desktop School ALNG LicSAPk MVL.
 - a. Office ProPlus All Lng Lic/SA Pack MVL Partners in Learning (лицензия на пакет Office Professional Plus)

- b. Windows 8
2. Система тестирования INDIGO.
3. Adobe Acrobat – свободно-распространяемое ПО
4. Интернет-браузеры Google Chrome, Firefox – свободно-распространяемое ПО
5. Консультант + версия проф.- справочная правовая система

Каждый обучающийся в течение всего обучения обеспечивается индивидуальным неограниченным доступом электронно-библиотечной системе и электронной информационно-образовательной среде.

14. Описание материально–технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Образовательный процесс обеспечивается специальными помещениями, которые представляют собой аудитории для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, помещения для самостоятельной работы студентов и помещения для хранения и профилактического обслуживания учебного оборудования.

Специальные помещения соответствуют действующим противопожарным правилам и нормам, укомплектованы специализированной мебелью.

Аудитории лекционного типа, оснащенные проекционным оборудованием и техническими средствами обучения, обеспечивающими представление учебной информации большой аудитории, демонстрационным оборудованием.

Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой, обеспечивающей доступ к сети Интернет и электронной информационно-образовательной среде университета.

15. Методические указания для обучающихся по освоению дисциплины

Дисциплина «Программирование» состоит из 6 тем и изучается на лекциях, лабораторных занятиях и при самостоятельной работе обучающихся. Обучающийся для полного освоения материала должен не пропускать занятия и активно участвовать в учебном процессе. Кроме того, обучающиеся должны ознакомиться с программой дисциплины и списком основной и дополнительной рекомендуемой литературы.

Основной теоретический материал дается на лекционных занятиях. Лекции включают все темы и основные вопросы теории и практики дисциплины «Базы данных». Для максимальной эффективности изучения необходимо постоянно вести конспект лекций, знать рекомендуемую преподавателем основную и дополнительную учебную литературу,

позволяющую дополнить знания и лучше подготовиться к лабораторным занятиям.

Для закрепления теоретического материала проводятся лабораторные занятия. В ходе занятий разбираются основные и дополнительные теоретические вопросы информационных систем, решаются практические задачи на разработку и обоснование стратегических решений, проводятся тестирования по результатам изучения тем.

На изучение каждой темы выделено в соответствии с рабочей программой дисциплины количество часов лабораторных занятий, которые проводятся в соответствии с вопросами, рекомендованными к изучению по определенным темам. Обучающиеся должны регулярно готовиться к лабораторным занятиям. При подготовке к занятиям следует руководствоваться конспектом лекций и рекомендованной литературой.

Для эффективного освоения материала дисциплины учебным планом предусмотрена самостоятельная работа, которая должна выполняться в обязательном порядке. Выполнение самостоятельной работы по темам дисциплины, позволяет регулярно проводить самооценку качества усвоения материалов дисциплины и выявлять аспекты, требующие более детального изучения. Задания для самостоятельной работы предложены по каждой из изучаемых тем и должны готовиться индивидуально и к указанному сроку. По необходимости студент может обращаться за консультацией к преподавателю. Выполнение заданий контролируется и оценивается преподавателем.

В случае посещения обучающегося лекций и лабораторных занятий, изучения рекомендованной основной и дополнительной учебной литературы, а также своевременного и самостоятельного выполнения заданий, подготовка к экзамену по дисциплине сводится к дальнейшей систематизации полученных знаний, умений и навыков.

16. Методические рекомендации по организации изучения дисциплины для преподавателей, образовательные технологии

Оценочные средства для контроля успеваемости и результатов освоения дисциплины (модуля):

- а) для текущей успеваемости: практические задачи, самостоятельная работа, контрольная работа;
- б) для самоконтроля обучающихся: тесты;
- в) для промежуточной аттестации: вопросы для экзамена, практические задания

При реализации различных видов учебной работы по дисциплине «Программирование», оценка возможных последствий и контроль над исполнением» используются следующие образовательные технологии:

- 1) лекции с использованием методов проблемного изложения материала;
- 2) обсуждение в группе (обсуждение подготовленных студентами

вопросов для самостоятельного изучения)

№	Занятие в интерактивной форме	Количество часов по очной форме		Количество часов по заочной форме	
		Лекция	Лаборат.	Лекция	Лаборат.
2	Введение в программирование на процедурном языке Лекция с демонстрацией видеоматериалов (слайды) Лабораторные занятия с применением следующих технологий: - обсуждение в группе (обсуждение подготовленных студентами вопросов для самостоятельного изучения)	1	1	0,5	1
3	Описание и обработка данных структурированного типа Лекция с демонстрацией видеоматериалов (слайды) Лабораторные занятия с применением следующих технологий: - обсуждение в группе (обсуждение подготовленных студентами вопросов для самостоятельного изучения)	2	2	0,5	2
4	Модульное программирование Лекция с демонстрацией видеоматериалов (слайды) Лабораторные занятия с применением следующих технологий: - обсуждение в группе (обсуждение подготовленных студентами вопросов для самостоятельного изучения)	1	1	1	2
7	Итого	4	4	2	5

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ (МОДУЛЮ)**

ПРОГРАММИРОВАНИЕ

Направление подготовки: 38.03.05 Бизнес-информатика
Направленность: «Электронный бизнес»

1. Паспорт фонда оценочных средств

1.1 Компетенции, формируемые в процессе изучения дисциплины:

Индекс	Формулировка компетенции
ОПК-3	способностью работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях
ПК-18	способность использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования

1.2. Сведения об иных дисциплинах (преподаваемых, в том числе, на других кафедрах) участвующих в формировании данных компетенций:

1.2.1. Компетенция ОПК-3 формируется в процессе изучения дисциплин (прохождения практик):

Теоретические основы информатики
Вычислительные системы, сети, телекоммуникации
Объектно-ориентированный анализ и программирование
Информационные системы управления производственной компанией
Управление ИТ-сервисами и контентом
Компьютерная графика и дизайн
Учебная практика. Практика по получению первичных профессиональных умений и навыков проектной деятельности, в том числе первичных умений и навыков научно-исследовательской деятельности

1.2.2. Компетенция ПК-18 формируется в процессе изучения дисциплин (прохождения практик):

Линейная алгебра
Математический анализ
Теория вероятностей и математическая статистика
Дискретная математика
Объектно-ориентированный анализ и программирование
Информационные технологии анализа и оценки электронного бизнеса
Производственная практика. Преддипломная практика
Учебная практика. Практика по получению первичных профессиональных умений и навыков проектной деятельности, в том числе первичных умений и навыков научно-исследовательской деятельности

1.3. Этапы формирования и программа оценивания контролируемой компетенции:

№	Код контролируемой компетенции (или ее части)	Контролируемые модули, разделы (темы) дисциплины*	Наименование оценочного средства**
1 2	ОПК-3 ПК-18	1. Основы алгоритмизации 2. Введение в программирование на процедурном языке 3. Описание и обработка данных структурированного типа 4. Модульное программирование 5. Динамические переменные и указатели 6. Работа с файлами	Практические задачи Самостоятельная работа Контрольная работа

Процедура оценивания

1. Процедура оценивания результатов освоения программы учебной дисциплины включает в себя оценку уровня сформированности компетенций студента при осуществлении текущего контроля и проведении промежуточной аттестации.

2. Уровень сформированности компетенции (ОПК-3, ПК-18) определяется по качеству выполненной студентом работы и отражается в следующих формулировках: высокий, хороший, достаточный, недостаточный.

3. При выполнении студентами заданий текущего контроля и промежуточной аттестации оценивается уровень обученности «знать», «уметь», «владеть» в соответствии с запланированными результатами обучения и содержанием рабочей программы дисциплины:

– профессиональные знания студента могут проверяться при ответе на теоретические вопросы, выполнении тестовых заданий, практических работ,

– степень владения профессиональными умениями – при решении ситуационных задач, выполнении практических работ и других заданий.

4. Результаты выполнения заданий фиксируются в баллах в соответствии с показателями и критериями оценивания компетенций. Общее количество баллов складывается из:

– суммы баллов за выполнение практических заданий на выявление уровня обученности «уметь»,

– суммы баллов за выполнение практических заданий на выявление уровня обученности «владеть»,

– суммы баллов за ответы на теоретические и дополнительные вопросы,

5. По итогам текущего контроля и промежуточной аттестации в соответствии с показателями и критериями оценивания компетенций

определяется уровень сформированности компетенций студента и выставляется оценка по шкале оценивания.

1.4. Показатели и критерии оценивания компетенций, шкала оценивания

Компетенции	Показатели оценивания	Критерии оценивания компетенций				Итого:				
		Высокий (верно и в полном объеме) 5 б.	Средний (с незначительными замечаниями) 4 б.	Низкий (на базовом уровне, с ошибками) 3 б.	Недостаточный (содержит большое количество ошибок/ответ не дан) – 2 б.					
Теоретические показатели										
ОПК-3 ПК-18	Знает основные технологии программирования	Верно, и в полном объеме знает основные технологии программирования	С незначительными замечаниями знает основные технологии программирования	На базовом уровне, с ошибками знает основные технологии программирования	Не знает основные технологии программирования	5				
	<i>Практические задачи</i>									
Практические показатели										
ОПК-3 ПК-18	Умеет проектировать, внедрять и эксплуатацию ИС и ИКТ	Верно, и в полном объеме может проектировать, внедрять и эксплуатацию ИС и ИКТ	С незначительными замечаниями может проектировать, внедрять и эксплуатацию ИС и ИКТ	На базовом уровне, с ошибками может проектировать, внедрять и эксплуатацию ИС и ИКТ	Не может проектировать, внедрять и эксплуатацию ИС и ИКТ	20				
	<i>Самостоятельная работа</i>									
	Умеет разрабатывать алгоритмы решения.						Верно, и в полном объеме может разрабатывать алгоритмы решения.	С незначительными замечаниями может разрабатывать алгоритмы решения.	На базовом уровне, с ошибками может разрабатывать алгоритмы решения.	Не может разрабатывать алгоритмы решения.
	<i>Самостоятельная работа</i>									
Умеет программировать задачи обработки данных в предметной области	Верно, и в полном объеме может программировать задачи обработки данных в предметной области	С незначительными замечаниями может программировать задачи обработки данных в предметной области	На базовом уровне, с ошибками может программировать задачи обработки данных в предметной области	Не может программировать задачи обработки данных в предметной области						
<i>Самостоятельная работа</i>										
	Умеет выполнять тестирование и отладку программ	Верно, и в полном объеме может выполнять тестирование и отладку программ	С незначительными замечаниями может выполнять тестирование и отладку программ	На базовом уровне, с ошибками может выполнять тестирование и отладку программ	Не может выполнять тестирование и отладку программ					
	<i>Самостоятельная работа</i>									

Компетенции	Показатели оценивания	Критерии оценивания компетенций				Итого:
		Высокий (верно и в полном объеме) 5 б.	Средний (с незначительными замечаниями) 4 б.	Низкий (на базовом уровне, с ошибками) 3 б.	Недостаточный (содержит большое количество ошибок/ответ не дан) – 2 б.	
Владеет						
ОПК-3 ПК-18	Владеет методами и инструментальными средствами разработки программ. <hr/> <i>Контрольная работа</i>	Верно, и в полном объеме владеет методами и инструментальными средствами разработки программ.	С незначительными замечаниями владеет методами и инструментальными средствами разработки программ.	На базовом уровне, с ошибками владеет методами и инструментальными средствами разработки программ.	Не владеет методами и инструментальными средствами разработки программ.	10
	Владеет методами оформления программной документации <hr/> <i>Контрольная работа</i>	Верно, и в полном объеме владеет методами оформления программной документации	С незначительными замечаниями владеет методами оформления программной документации	На базовом уровне, с ошибками владеет методами оформления программной документации	Не владеет методами оформления программной документации	
Всего:						35

Шкала оценивания:*

Оценка	Баллы	Уровень сформированности компетенции
отлично	30-35	высокий
хорошо	25-29	хороший
удовлетворительно	18-24	достаточный
неудовлетворительно	17 и менее	недостаточный

2. Типовые контрольные задания для оценки результатов обучения по дисциплине и иные материалы для подготовки к промежуточной аттестации

2.1 Материалы для подготовки к промежуточной аттестации

Вопросы к экзамену:

1. Этапы решения задач на ЭВМ.
2. Алгоритм. Привести примеры алгоритмов (из жизни, из алгебры, геометрии, русского языка и т.д.).
3. Сформулировать свойства алгоритма. Привести примеры.
4. Понятие исполнителя. Система команд исполнителя. Примеры.
5. Рассказать о формальном исполнении алгоритма. Трассировка. Примеры.
6. Основные алгоритмические структуры. Примеры алгоритмов.
7. Полная и неполная развилка. Примеры.
8. Циклы. Примеры.
9. Аргументы, результаты, промежуточные величины. Присваивание значений. Примеры.
10. Отношение между величинами в качестве условий. Примеры.
11. Принципы структурного программирования.
12. Типы величин (целые, вещественные, логические, символьные). Диапазон допустимых значений, представление в памяти ЭВМ, операции, определенные над величинами каждого типа.
13. Вспомогательный алгоритм. Формальные и фактические параметры, соответствие между ними.
15. Процедурные и непроцедурные языки программирования (обзор).
16. Алфавит. Структура программы
17. Простые типы данных
18. Переменные и константы. Описание. Типизированные константы. Примеры.
19. Встроенные математические функции. Алгебраические выражения и правила их записи.
20. Логические выражения. Правила их записи и вычисления.
21. Оператор присваивания. Совместимость типов.
22. Процедуры ввода и вывода в консольном режиме. Форматированный вывод.
23. Линейные алгоритмы. Примеры.
24. Условный оператор. Примеры.
25. Оператор выбора. Примеры.
26. Перечислимый и диапазонный типы данных. Примеры.
27. Циклы с предусловием и с постусловием. Примеры.
28. Цикл с параметром. Преобразование цикла с параметром к циклу с пред- или постусловием. Примеры.

29. Программирование вложенных циклов. Примеры.
30. Одномерные массивы. Примеры задач.
31. Двумерные массивы. Примеры задач.
32. Принципы структурного программирования. Блочный (модульный) подход к построению алгоритмов решения задач.
33. Процедуры. Вызов процедуры в программах. Формальные и фактические параметры. Соответствие между параметрами. Примеры.
34. Параметры-переменные, параметры-значения, параметры-константы. Примеры.
35. Глобальные и локальные идентификаторы. Видимость объектов (идентификаторов). Примеры.
36. Область действия переменных и других идентификаторов.
37. Функции. Их отличие от процедур. Способ обращения к функции. Примеры.
38. Символьный тип данных. Строковый тип данных. Строковые переменные и константы. Двойственный характер строк.
39. Функции, ориентированные на работу со строками. Примеры.
40. Процедуры, ориентированные на работу со строками. Примеры.
41. "Пузырьковая" сортировка (обменами).
42. Сортировка выбором.
43. Сортировка вставками.
44. Рекурсия. Механизм рекурсии: действия на рекурсивном спуске и рекурсивном возврате. Понятие о глубине рекурсии. Примеры.
45. Примеры рекурсивных функций: количество цифр в записи данного натурального числа; поиск минимального элемента в массиве; проверка, является ли число простым.
46. Внешние подпрограммы.
47. Комбинированный тип данных (запись). Способы описания (примеры).
48. Способы обращения к компонентам записи.
49. Оператор присоединения и его использование.
50. Записи с вариантами. Примеры.
51. Понятие множества. Обозначение множества в Pascal.
52. Задание множественного типа и множественная переменная.
53. Операции над множествами. Примеры.
54. Понятие файла. Логические и физические файлы.
55. Файловые типы Pascal (общие понятия).
56. Понятие текущей длины файла, указателя или окна файла, буфера файла.
57. Общие процедуры и функции для работы с файлами.
58. Компонентные (типизированные) файлы: описание, отличие от других видов файлов.
59. Текстовые файлы, принципы работы. Процедуры и функции, ориентированные на работу с текстовыми файлами, ввод-вывод информации.

60. Нетипизированные файлы и процедуры ввода-вывода. Примеры использования.

61. Прямой и последовательный доступ к компонентам файлов.

62. Работа с файлами записей. Примеры.

63. Понятие модуля. Разделы модуля. Подключение модуля. Компиляция модулей.

64. Построение диаграмм.

65. Построение графиков функций.

Типовые контрольные задания:

1. Дано натуральное число n ($n > 1$). Получить его каноническое разложение на простые сомножители, то есть представить в виде произведения простых сомножителей. При этом в разложении допустимо указывать множитель 1. Например, $264 = 2 * 2 * 2 * 3 * 11$ (программе допустимо выдать ответ $264 = 1 * 2 * 2 * 2 * 3 * 11$).

2. Дана последовательность натуральных чисел, ограниченная вводом нуля. Проверить, начинается ли каждый из ее членов (со второго) с десятичной цифры, на которую оканчивается предыдущий. Например, таковой последовательностью будет являться 14 47 712 2179 9 9 93 0 (также сохранен ограничивающий ноль)

3. Определение принадлежности хотя бы одной точки заданного множества точек на плоскости внутренней области круга с центром в точке (a, b) и радиусом R

4. Дана строка символов. Заменить все маленькие английские буквы в строке соответствующими большими

5. Дана строка символов. Найти сумму записанных в строке чисел

6. Написать программу удаления из введенной строки s , первых n символов.

Написать программу поиск в массиве персон по фамилии

Написать программу сортировки массива персон по возрасту

7. Определение наличия среди элементов главной диагонали заданной целочисленной матрицы $A(N, N)$ хотя бы одного положительного нечетного элемент

8. Задача о Ханойских башнях. Формулировка задачи: Дано три стержня. На первом стержне размещены p дисков разных диаметров в порядке их уменьшения, так что сверху находится диск с наименьшим диаметром. Требуется переложить диски на третий стержень, соблюдая следующие правила: • можно перемещать лишь по одному диску; • больший диск не разрешается класть на меньший; • откладывать диски в сторону не разрешается.

9. Составить программу перевода двоичного числа в десятичное.

10. Рекурсивные алгоритмы: нахождения НОД и НОК двух чисел.

11. Преобразование слова лево в слово компот.

Образцы тестовых заданий для проведения текущего контроля и промежуточной аттестации по итогам освоения дисциплины, а также для контроля самостоятельной работы:

1. 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;
```

- а) определяет количество цифр в строке;
- б) подсчитывает количество нулей в строке;
- в) определяет сумму номеров позиций в строке, где стоят цифры;
- г) подсчитывает сумму цифр в строке;
- д) определяет сумму номеров позиций в строке, где стоят нули.

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

1) *i* := *n*; While (*i* >= 1) And (*m*[*i*] > 0) Do Dec (*i*); If *i* < 1 Then WriteLn ('*i* = 0') Else WriteLn ('*i* = ', *i*);

2) *k* := 0; For *i* := 1 To *n* Do If *m*[*i*] < 0 Then *k* := *i*; WriteLn ('*i* = ', *k*);

3) *i* := *n*; Repeat *i* := *i* – 1 Until (*m*[*i*] < 0); WriteLn ('*i* = ', *i*);

а) 1, 2;

б) 2, 3;

в) 1, 2, 3;

г) 2;

д) ни один из ответов 1–3 не верен.

3. 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;
```

а) к элементам столбцов в первой половине матрицы прибавляет элементы первого столбца соответствующей строки;

б) добавляет к матрице еще *M* столбцов с элементами, равными соответствующим элементам первого столбца;

в) к элементам четных столбцов прибавляет элементы первого столбца соответствующей строки;

г) к элементам четных строк прибавляет элементы первой строки соответствующего столбца;

д) меняет порядок столбцов таблицы.

```

4. 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;

```

возвращает значение

- а) True, если все элементы массива ненулевые;
- б) True, если в массиве есть элемент, равный нулю;
- в) False, если в массиве есть элемент, равный нулю;
- г) Not t;
- д) ни один из ответов 1–4 не верен.

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

- 1) соответствие по типу параметров;
- 2) соответствие по количеству параметров;
- 3) соответствие по типу используемых вспомогательных переменных;
- 4) соответствие по порядку перечисления;

лишним является:

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

Литература для подготовки к экзамену:

а) нормативные документы

1. Об информации, информационных технологиях и защите информации: Федер. закон от 27.07.2006 N 149-ФЗ (ред. от 24.11.2014) . - КонсультантПлюс. [Электрон. ресурс]. – Электрон. дан. – [М., 2014].

2. Об электронной цифровой подписи: Федер. закон от 06.04.2011 N 63-ФЗ (ред. от 28.06.2014). - КонсультантПлюс. [Электрон. ресурс]. – Электрон. дан. – [М., 2014].

3. О персональных данных: Федер. закон от 27.07.2006 N 152-ФЗ (ред. от 21.07.2014) . - КонсультантПлюс. [Электрон. ресурс]. – Электрон. дан. – [М., 2014].

б) основная литература

1. Математическое программирование: Учебник / Балдин К.В.,

Брызгалов Н.А., Рукосуев А.В., - 2-е изд. - М.: Дашков и К, 2018. - 218 с. -
Режим доступа: <http://znanium.com/catalog/product/415097>

2. Программирование в среде Lazarus для школьников и студентов:
Учебное пособие / Гуриков С.Р. - М.: Форум, НИЦ ИНФРА-М, 2016. - 336 с. -
Режим доступа: <http://znanium.com/catalog/product/520628>

в) дополнительная литература

1. Программирование на языке Си: Учебное пособие /А.В. Кузин, Е.В. Чумакова - М.: Форум, НИЦ ИНФРА-М, 2015. - 144 с. - Режим доступа:
<http://znanium.com/catalog/product/505194>

Промежуточная аттестация

2.2. Комплект экзаменационных билетов для проведения промежуточной аттестации

АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность: «Электронный бизнес»

Дисциплина: «Программирование»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 1

1. Основные приемы работы с массивами. Описание, ввод и вывод элементов. Алгоритмы обработки массивов: линейный поиск, инверсия, сортировка и сжатие.
2. Найти произведение $P=(2-1)^2(3-1)^3(4-1)^4 \dots (n-1)^n$
3. Дан массив вещественных чисел. Определить сколько элементов массива превосходят по модулю заданное число a .

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 2

1. Структура и виды подпрограмм. Механизмы передачи параметров. Процедурный и функциональный типы. Примеры использования
2. Найти произведение $P=2(2-1)(3-1)(4-1) \dots (n-1)$
3. Дан массив вещественных чисел $B[20]$. Найти сумму минимального и максимального элементов массива.

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 3

1. Рекурсия. Примеры рекурсивных алгоритмов и их программной реализации.
2. Найти сумму $(a-1)^2+(a-2)^3+(a-3)^4+\dots+(a-n)^{n+1}$
3. Дан массив $F[15]$ ненулевых чисел. Разделить каждый элемент массива на 5й элемент, возведённый в квадрат

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ № 4

1. Общая структура модуля и назначение отдельных разделов. Пример модуля.
2. Найти сумму $7+(a-1)^1+(a-2)^2+(a-3)^3+\dots+(a-n)^n$
3. Дана символьная константа «тригонометрия». Составить из этого слова три новых слова, используя при этом все три символьные функции и операцию слияния.

Промежуточная аттестация
Комплект тестовых заданий для проведения экзамена по дисциплине

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность: «Электронный бизнес»

Дисциплина: «Программирование»

Тестовые задания для проверки уровня обученности **ЗНАТЬ:**

1. Определение «Однозначность выполнения каждого отдельного шага преобразования информации» относится к свойству алгоритмов, которое называется...

- а) дискретность
- б) массовость
- в) выполнимость
- г) дискретность

2. Перевод программ с языка высокого уровня на язык более низкого уровня обеспечивает программа -

- а) ассемблер
- б) паскаль
- в) компилятор
- г) фортран

3. Разработке алгоритма предшествует

а) постановка задачи, разработка математической модели, проектирование программ

б) постановка задачи, разработка математической модели

в) постановка задачи, выбор метода решения, проектирование программ

г) постановка задачи, разработка математической модели, выбор метода решения

Тестовые задания для проверки уровня обученности **УМЕТЬ, ВЛАДЕТЬ:**

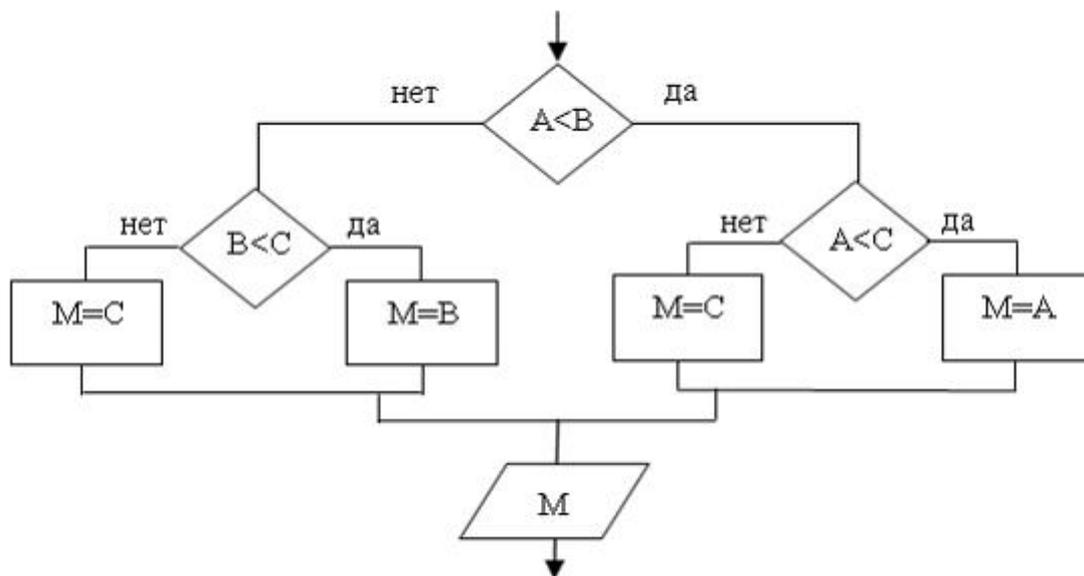
1. В программе, вычисляющей произведение отрицательных чисел из N введенных с клавиатуры,

ВВОД N
 P:=1
НЦ для i от 1 до N
ВВОД a
ЕСЛИ
ТО P:=P*a
ВСЕ
КЦ
ВЫВОД P

необходимо вставить условие...

- а) $a=1$
- б) $a>0$
- в) $a<0$
- г) $a<N$

2. Изображенный на рисунке фрагмент алгоритма



определяет...

- а) минимум из трех чисел
- б) наибольшее из чисел B и C
- в) наименьшее из чисел A и B
- г) максимум из трех чисел

2.3. Критерии оценки для проведения экзамена по дисциплине

После завершения тестирования на экзамене на мониторе компьютера высвечивается результат – процент правильных ответов. Результат переводится в баллы и суммируется с текущими семестровыми баллами.

Максимальная сумма (100 баллов), набираемая студентом по дисциплине, предусматривающей в качестве формы промежуточной аттестации экзамен, включают две составляющие.

Первая составляющая – оценка регулярности и своевременности качества выполнения студентом учебной работы по изучению дисциплины в течение семестра (сумма не более 60 баллов).

Вторая составляющая – оценка знаний студента на экзамене (не более 40 баллов).

Перевод полученных итоговых баллов в оценки осуществляется по следующей шкале:

- с 86 до 100 баллов – «отлично»;
- с 71 до 85 баллов – «хорошо»;
- с 50 до 70 баллов – «удовлетворительно»

Если студент при тестировании отвечает правильно менее, чем на 50 %, то автоматически выставляется оценка «неудовлетворительно» (без суммирования текущих рейтинговых баллов), а студенту назначается переэкзаменовка в дополнительную сессию.

2.4. Методические материалы, определяющие процедуру оценивания по дисциплине

Общая процедура оценивания определена Положением о фондах оценочных средств.

1. Процедура оценивания результатов освоения программы дисциплины включает в себя оценку уровня сформированности общекультурных и профессиональных компетенций студента, уровней обученности: «знать», «уметь», «владеть».

2. При сдаче экзамена:

– профессиональные знания студента могут проверяться при ответе на теоретические вопросы, при выполнении тестовых заданий, практических работ;

– степень владения профессиональными умениями, уровень сформированности компетенций (элементов компетенций) – при решении ситуационных задач, выполнении практических работ и других заданий.

3. Результаты промежуточной аттестации фиксируются в баллах. Общее количество баллов складывается из следующего:

- до 60% от общей оценки за выполнение практических заданий,
- до 30% оценки за ответы на теоретические вопросы,
- до 10% оценки за ответы на дополнительные вопросы.

**АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ТЕКУЩЕГО КОНТРОЛЯ И ТЕКУЩЕЙ АТТЕСТАЦИИ ПО
ДИСЦИПЛИНЕ (МОДУЛЮ)**

ПРОГРАММИРОВАНИЕ

Направление подготовки: 38.03.05 Бизнес-информатика
Направленность: «Электронный бизнес»

1. Материалы для текущего контроля

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

Комплект практических задач.

по дисциплине «Программирование»

Вариант № 1

1. Составить линейный алгоритм: расстояние между автомобилями — S км. Определить расстояние между ними через t ч, если первый движется со скоростью v_1 , а второй — v_2 км/ч. Примечание. Рассмотреть случаи движения навстречу друг другу и в противоположные стороны друг от друга.

2. Составить алгоритм с развилкой: выяснить, что больше: площадь правильного шестиугольника со стороной a или прямоугольника с измерениями b, c .

3. Составить алгоритм с использованием цикла: найти сумму членов ряда, для которых выполняется неравенство $|a_i| \geq \square$:

$$a_i = \sum_{i=0}^n (-1)^i \frac{x^{2i+1}}{(2i+1)!}$$

4. Составить алгоритм решения задачи целочисленной арифметики: вычеркнуть из записи данного натурального числа N первую цифру.

5. Составить алгоритм решения задачи с использованием вспомогательных алгоритмов: поменять порядок следования цифр в заданном натуральном числе N на обратный, удалив при этом из записи получаемого числа все вхождения цифры k .

Вариант №2

1. Изобразить на плоскости (X, Y) область, в которой и только в которой истинно указанное выражение

$(X \leq 0) \text{ and } (\text{Sqr}(X) + \text{Sqr}(Y) \leq 4) \text{ or } (X \geq 0) \text{ and } (Y \leq -X + 2) \text{ and } (Y \geq X - 2)$

2. Объяснить, в чем заключается синтаксическая ошибка (или ошибки) в приведенной программе. Написать тот вариант программы, который, по Вашему мнению, будет правильным.

```
Program My_Error;  
Var X, Y, Сумма: Integer;
```

```

Begin
    WriteLn( Введите значение X); ReadLn(X);
    WriteLn( Введите значение Y); ReadLn(Y);
    Сумма = X + Y;
    WriteLn(Сумма)

```

End.

3. Составить программу идентификации треугольника по сторонам a , b , c . Определяемое свойство: является прямоугольным или не является прямоугольным (удобно использовать сравнение квадрата одной из сторон с суммой квадратов двух других сторон).

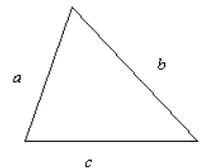
4. Решить задачу с использованием оператора выбора: по введенному числу грибов k напечатать фразу “Мы нашли в лесу k грибов” и согласовать при этом окончание слова “гриб” с числом k . (Количество грибов может быть любым целым числом: 1, 3, 34, 127 и т.п. Окончание фразы определяется значением последней цифры.)

5. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```

Program Pr;
Var K, I, D, S: Integer;
Begin WriteLn(‘Введите значение K: ’); ReadLn(K);
    For I := 100 To 300 Do
        Begin D := 2;
            S := 0;
            While D <= I div 2 + 1 Do
                Begin If I mod D = 0 Then S := S + D;
                    D := D + 1
                End;
            If S = K Then WriteLn(I);
        End
    End

```



End

End.

6. Решить задачу с использованием цикла: найти k -е простое число в арифметической прогрессии 11, 21, 31, 41, 51, 61, ...

7. Определить, что будет напечатано программой по ходу ее выполнения:

```

Program Schet;
Var M, N: Word;
Begin M := 0; N := 0;
    Repeat M := M + 1;
        N := N + M;
        Write(M:5, N:3)
    Until N mod 7 = 0;
    Writeln;
    M := 0; N := 0;
    While M + N < 20 Do
        Begin M := M + 1; N := N + M; Write(M:5, N:3) End

```

End.

8. Решить задачу с использованием подпрограммы: вывести наибольшую из первых цифр трех заданных чисел. Например, если $a = 25$, $b = 730$, $c = 1995$, то надо вывести цифру 7.

9. В данной последовательности a_1, a_2, \dots, a_n определить максимальное число среди элементов с номерами, кратными числу k . Например, для последовательности $-1, 0, 12, -77, 22, -6, 70, 11, 3$ и $k = 3$ получаем ответ 12.

10. Вычислить
$$\sqrt{2 + \sqrt{2 + \dots + \sqrt{2}}}$$

(n слагаемых).

Вариант №3

1. Определить, какой объём оперативной памяти требуется для хранения каждой из объявленных переменных. Указать, какую из переменных не удастся разместить в памяти из-за превышения допустимого объёма для статических переменных (возникнет ошибка Structure too large).

- Var A : Array[0..99] Of LongInt;
- Var B : Array[Integer] Of Real;
- Var C : Array[-100..100] Of Real;
- Var D : Array['a'..'z'] Of Extended;
- Var E : Array[1..100, 0..39] Of ShortInt;

2. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```
Program Kr_N_4;
Const NMax = 50; Type Mass = Array[0..NMax] Of Real;
Var X : Mass; I, N : Byte; Vsp : Real;
Begin Write('Сколько элементов в массиве? '); ReadLn(N); N := N - 1;
  Randomize;
  For I := 0 To N Do
    Begin X[i] := -50 + Random * 100.1; Write(X[i]:10:5) End;
  WriteLn; I := 0;
  While I < N Do
    Begin Vsp := X[i]; X[i] := X[i + 1];
      X[i + 1] := Vsp; I := I + 2
    End;
  For I := 0 To N Do Write(X[i]:10:5); WriteLn;
End.
```

3. Решить задачу: подсчитать количество элементов массива, каждый из которых не меньше всех своих предшественников (т.е. элементов с меньшими индексами). Например, для массива $-2, 3, 0, 13, 13, 4, -3$ ответ будет 4.

4. Решить задачу: заполнить квадратную матрицу порядка N по следующему правилу: элементы выше главной диагонали совпадают с разностью первого и второго индексов, выше побочной диагонали — с

произведением индексов. Примечание. К элементам, попадающим в пересечение областей, применить следующее правило: они получаются, как сумма значений, полученных по первому и второму правилу. Все остальные элементы матрицы равны нулю.

5. *Решить задачу:* в целочисленной прямоугольной таблице размером $M \times N$ определить наибольшее число, кратное заданному K , или сообщить об отсутствии таких чисел.

6. Точно и четко сформулировать условие задачи, которая решается в данной программе:

```

Program Kr_4_z2;
Var S : String; M, I, N, J : Byte; Vsp : Char;
Begin
  Write('Введите строку: '); ReadLn(S); N := Length(S);
  For I := 1 To N - 1 Do
  Begin
    m := I;
    For J := I + 1 To N Do If S[j] < S[m] Then m := j;
    Vsp := S[i]; S[i] := S[m]; S[m] := Vsp
  End;
  WriteLn('Ответ: ', S)
End.

```

6. *Решить задачу:* в строке заменить указанный символ C на символ $C1$. Подсчитать число замен.

Вариант №4

1. Определить, какой минимальный объём оперативной памяти требуется для хранения каждой из объявленных переменных.

- Type CompCh = Record re, im: Real end;
Var Spis : array[1..100,0..39] of CompCh;
- Type Kol = Record name : string[10]; fone : LongInt end;
Var T: array [-1..10] of Kol;
- Var Uch: Record fam, name: string[15]; year : Word end;
- Var M: set of '0'..'9';

2. Описать рекурсивную функцию $pow(x, n)$ от вещественного x ($x \neq 0$) и целого n , которая вычисляет величину x^n согласно формуле

$$x^n = \begin{cases} 1, & x = 0; \\ \frac{1}{x^{|n|}}, & x < 0; \\ x \cdot x^{n-1}, & x > 0. \end{cases}$$

3. Вычислить значения выражений

- [2,3,5,7]<=[1..9]; b) ['c', 'b']=['b'..'c'];
- trunc(3.9) in [1,3,5]; d) [2..13]*[3,13..60]+[4..10]-[5..15]*[6];
- [2..10]-[4,6]-[2..12]*[8..15]; f) (('0'..'7')+['2'..'9'])*(['a']+['z']).

4. Решить задачу с использованием множества: имеется строка, содержащая русский текст. Вывести те буквы, которые являются общими для первых трёх предложений текста.

5. Заданы некоторые типы данных и объявлены переменные:

```
Type Complex = Record re, im: Real end;
```

```
Point = Record x, y: Real end;
```

```
Var z, w: Complex; p: Point; re: Real;
```

Определить, какие значения будут иметь переменные $z.re$, $z.im$, $w.re$, $w.im$, $p.x$, $p.y$, re после выполнения следующих операторов:

```
with z do begin re := 0; im := 1 end;
```

```
w := z; re := 2;
```

```
with z do re := 1;
```

```
with z, w do im := -im;
```

```
with p do begin x := re; y := 2 end;
```

6. Сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в массиве:

```
Type Stud = Record
```

```
fam : string[12];
```

```
num : 121..126;
```

```
otm1, otm2, otm3 : 2..5;
```

```
End;
```

```
Var Kurs : array [1..100] of Stud;
```

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости

Примерный вариант контрольной работы №5 (файлы, модули)

1. Точно и однозначно сформулировать условие задачи, решение которой приведено ниже.

```
Program Example_File_1;
```

```
Var F : File Of Real; G : File Of Integer;
```

```
S : String; A : Real; B : Integer;
```

```
Begin
```

```
Write('Имя файла? '); ReadLn(S); Assign(F, S);
```

```
Write('Имя файла? '); ReadLn(S); Assign(G, S);
```

```
ReSet(F); ReWrite(G);
```

```
While Not Eof(F) Do
```

```
Begin
```

```
Read(F, A); B := Trunc(A); B := Ord(Odd(B)) * B; Write(G, B)
```

```
End; Close(F); Close(G)
```

```
End.
```

2. Решить задачу: заполнить файл целыми числами из отрезка $[-1000; 1000]$ с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.

3. *Решить задачу:* сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в файле:

```
Type Stud = Record
fam: string[12];
num:121..126;
otm1, otm2, otm3: 2..5;
end;
Var Kurs: File of Stud;
```

Написать программу, которая вводит эту информацию и печатает номера групп в порядке убывания средней успеваемости их студентов.

4. *Решить задачу:* удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.

5. *Решить задачу:* разработать модуль для работы с векторами вида $a(a_1, a_2)$. Реализовать операции сложения, вычитания, нахождения длины вектора. Используя модуль, переписать из данного файла в другой те векторы, длины которых больше заданного K .

Примерный вариант контрольной работы №6 (файлы, модули)

1. Точно и однозначно сформулировать условие задачи, решение которой приведено ниже.

```
Program Example_File_1;
Var F : File Of Real; G : File Of Integer;
S : String; A : Real; B : Integer;
Begin
Write('Имя файла? '); ReadLn(S); Assign(F, S);
Write('Имя файла? '); ReadLn(S); Assign(G, S);
ReSet(F); ReWrite(G);
While Not Eof(F) Do
Begin
Read(F, A); B := Trunc(A); B := Ord(Odd(B)) * B; Write(G, B)
End; Close(F); Close(G)
End.
```

2. *Решить задачу:* заполнить файл целыми числами из отрезка $[-1000; 1000]$ с помощью датчика случайных чисел. Переписать в один из новых файлов те из компонент исходного файла, модуль которых является простым числом, в другой — все остальные.

3. *Решить задачу:* сведения о результатах сессии студентов второго курса (фамилия, номер группы и оценки по трем экзаменам) хранятся в файле:

```
Type Stud = Record
fam: string[12];
num:121..126;
otm1, otm2, otm3: 2..5;
end;
```

Var Kurs: File of Stud;

Написать программу, которая вводит эту информацию и печатает номера групп в по-рядке убывания средней успеваемости их студентов.

4. *Решить задачу:* удалить из файла, содержащего целые числа, неположительные компоненты. Дополнительных файлов и массивов не использовать.

5. *Решить задачу:* разработать модуль для работы с векторами вида $a(a_1, a_2)$. Реализовать операции сложения, вычитания, нахождения длины вектора. Используя модуль, переписать из данного файла в другой те векторы, длины которых больше заданного K .

Критерии оценки:

- оценка «отлично» выставляется студенту, если контрольная работа выполнена верно и в полном объеме;
- оценка «хорошо» выставляется студенту, если контрольная работа выполнена с незначительными замечаниями;
- оценка «удовлетворительно» выставляется, если контрольная работа выполнена на базовом уровне, но с ошибками;
- оценка «неудовлетворительно» выставляется, если содержится большое количество ошибок, контрольная работа не выполнена.

2. Материалы для проведения текущей аттестации

Текущая аттестация 1

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

Комплект заданий для проведения текущей аттестации №1 (в форме самостоятельной работы)

по дисциплине «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 Sring[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

K := 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', 'п', 'р']
    Then Begin K := K + 1;
          Delete(S, I, 1); I := I - 1
    End;
    I := I + 1
  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); K := FileSize(F) - 1;
For I := 0 To K 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.

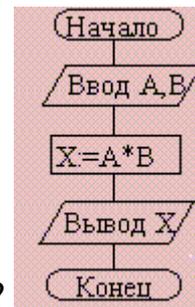
7. Имеется описание

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];
```

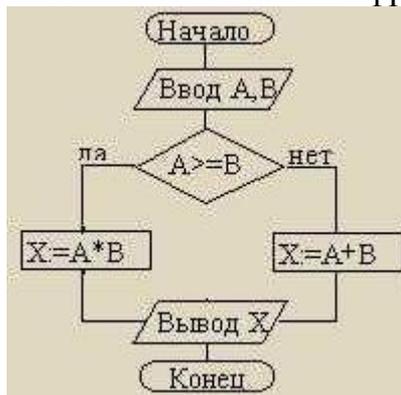
После исполнения этого фрагмента переменная V имеет значение

- 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.

Критерии оценки:

Студент аттестован, если правильно ответил более чем на 10 вопросов.

Текущая аттестация 2

АВТНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
ЦЕНТРОСОЮЗА РОССИЙСКОЙ ФЕДЕРАЦИИ
«РОССИЙСКИЙ УНИВЕРСИТЕТ КООПЕРАЦИИ»
КАЗАНСКИЙ КООПЕРАТИВНЫЙ ИНСТИТУТ (ФИЛИАЛ)

Комплект заданий для проведения текущей аттестации №2 (в форме контрольной работы)

по дисциплине «Программирование»

1. 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.

2. 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.

3. 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.

4. Кодировается шестнадцатибитовое целое со знаком (тип Integer).

1111111111110000₂ — это код числа

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

5. 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.

6. Из перечисленных ниже в программе обязательен

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

7. 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.

8. 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.

9. 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.

10. 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) $a = 7$, $b = 28$.

11. Во фрагмент алгоритма

For $K := 10$ To 99 Do

Begin $P1 := K \text{ Div } 10$;

$P2 := K \text{ Mod } 10$;

$S := P1 + P2$;

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) $(P1 = N) \text{ And } (P2 = N) \text{ And } (S = N)$.

12. Значения переменных p и d после выполнения фрагмента алгоритма
 $k := 47$; Case $k \text{ Mod } 9$ Of

5: Begin $d := k$; $p := \text{True}$ End;

0..2: Begin $d := 2$; $p := \text{False}$ End;

8: Begin $d := 1$; $p := \text{False}$ End

Else Begin $d := 1$; $p := \text{True}$ 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) $p = \text{True}$, $d = 2$.

13. 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.

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

а) $i := n$; While $(i \geq 1) \text{ And } (m[i] > 0)$ 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.

15. 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.

16. 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;

```

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.

17. 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.

Критерии оценки:

Студент аттестован, если правильно ответил более чем на 10 вопросов.