



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
филиал федерального государственного бюджетного образовательного учреждения
высшего образования «Кубанский государственный университет»
в г.Геленджике



УТВЕРЖАЮ
Профессор по работе с филиалами

А.А. Евдокимов

2023 г.

Рабочая программа дисциплины

ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

специальность 09.02.07 Информационные системы и программирование

2023

Рабочая программа учебной дисциплины ОП.03.Информационные технологии разработана на основе Федерального государственного образовательного стандарта (далее – ФГОС) по специальности среднего профессионального образования (далее СПО) 09.02.07 Информационные системы и программирование, утвержденного приказом Минобрнауки РФ от 09.12.2016 № 1547 (зарегистрирован в Минюсте России 26.12.2016 № 44936)

Дисциплина ОП.04 Основы алгоритмизации и программирования
Форма обучения очная
Учебный год 2023-2024

3 курс	6 семестр
лекции	76 час.
практические занятия	40 час.
лабораторные занятия	36 час.
самостоятельные занятия	2 час.
форма итогового контроля	экзамен

Составитель: преподаватель

Л.А. Благова

Утверждена на заседании предметной (цикловой) комиссии профессиональных дисциплин специальностей 09.02.03 Программирование в компьютерных системах и 09.02.07 Информационные системы и программирование
Протокол № 10 от «25» мая 2023 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальностей 09.02.03 Программирование в компьютерных системах и 09.02.07 Информационные системы и программирование

Л.А. Благова

подпись

Рецензенты:

Директор ООО «Современные
информационные технологии»



А.В.Сметанин

Системный администратор
ЗАО «Геленджикский Дельфинарий»





Г.П. Кривошеенко

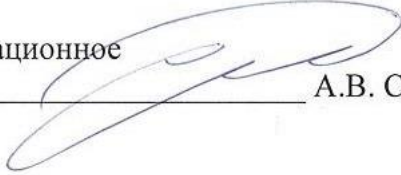
ЛИСТ
согласования рабочей учебной программы по дисциплине
ОП.04 Основы алгоритмизации и программирования

Специальность среднего профессионального образования:
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

Заместитель директора по УР филиала  Т.А. Резуненко

Заведующая сектором библиотеки филиала  Л.Г. Соколова

Инженер-электроник (программно-информационное
обеспечение образовательной программы)  А.В. Сметанин

СОДЕРЖАНИЕ

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ».....	5
1.1. Область применения программы.....	5
1.2. Место дисциплины в структуре основной профессиональной образовательной программы ...	5
1.3. Цель и планируемые результаты освоения дисциплины.....	5
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций).....	6
2 СТРУКТУРА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	14
1.1. Объем учебной дисциплины и виды учебной работы.....	14
2.2. Структура дисциплины.....	14
2.3. Тематический план и содержание учебной дисциплины.....	16
2.4. Содержание разделов дисциплины.....	21
2.4.2. Занятия семинарского типа.....	22
2.4.4. Содержание самостоятельной работы.....	24
3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	25
3.1. Образовательные технологии при проведении лекций.....	25
3.2. Образовательные технологии при проведении практических занятий (лабораторных работ).....	25
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	27
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.....	27
4.2. Информационное обеспечение реализации программы.....	27
5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ.....	28
5.1 Основная литература.....	28
5.2 Дополнительная литература.....	28
5.3 Периодические издания.....	28
5.4 Перечень ресурсов информационно-телекоммуникационной сети Интернет.....	29
6. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ.....	30
6.1. Критерии оценивания практических работ.....	30
6.2. Критерии оценивания конспектов.....	31
6.3. Критерии оценивания презентаций.....	31
6.4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ МОДУЛЯ.....	33
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ.....	38
7.1. Паспорт фонда оценочных средств.....	38
7.2. Критерии оценки знаний.....	38
7.3. Оценочные средств для проведения текущей аттестации.....	39
7.4. Оценочные средства для проведения промежуточной аттестации.....	39
7.4. Оценочные средства промежуточной аттестации.....	68
7.4.1. Примерные вопросы.....	68
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	81

1. ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

1.1. Область применения программы

Рабочая программа ОП.04 «Основы алгоритмизации и программирования» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07. «Информационные системы и программирование»

1.2. Место дисциплины в структуре основной профессиональной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу. Она обеспечивает профессиональный уровень подготовки специалиста и соответствует развитию их профессионально значимых качеств.

1.3. Цель и планируемые результаты освоения дисциплины

Для освоения дисциплины обучающиеся используют знания, умения и навыки, сформированные при изучении ЕН.02 «Дискретная математика с основами математической логики», УДВ.01 «Информатика», ОП.03 «Информационные технологии».

Код ПК, ОК	Умения	Знания	Практический опыт
ОК 1 ОК 2 ОК 4 ОК 5 ОК 9 ОК 10 ПК 1.1- ПК 1.5 ПК 2.4, 2.5	<p>Разрабатывать алгоритмы для конкретных задач.</p> <p>Использовать программы для графического отображения алгоритмов.</p> <p>Определять сложность работы алгоритмов.</p> <p>Работать в среде программирования.</p> <p>Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.</p> <p>Оформлять код программы в соответствии со стандартом кодирования.</p> <p>Выполнять проверку, отладку кода программы.</p>	<p>Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.</p> <p>Эволюцию языков программирования, их классификацию, понятие системы программирования.</p> <p>Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.</p> <p>Подпрограммы, составление библиотек подпрограмм</p> <p>Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения</p>	<p>Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.</p> <p>Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля</p> <p>Использовать инструментальные средства на этапе отладки программного продукта.</p> <p>Проводить тестирование программного модуля по определенному сценарию</p> <p>Проводить тестирование программного модуля по определенному сценарию.</p> <p>Использовать инструментальные средства на этапе тестирования программного продукта</p> <p>Анализировать алгоритмы, в том числе с применением инструментальных средств.</p> <p>Осуществлять рефакторинг и оптимизацию программного кода</p> <p>Разрабатывать тестовые наборы (пакеты) для программного модуля.</p> <p>Разрабатывать тестовые сценарии программного средства.</p> <p>Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования</p> <p>Инспектировать разработанные программные модули на предмет</p>

			соответствия стандартам кодирован.
--	--	--	------------------------------------

1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)

В результате освоения учебной дисциплины обучающийся должен уметь:

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования.

В результате освоения учебной дисциплины обучающийся должен знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языке.

В результате освоения учебной дисциплины обучающийся осваивает элементы компетенций:

Перечень профессиональных компетенций, элементы которых формируются в рамках учебной дисциплины

Код	Наименование профессиональных компетенций
ПК 1.1.	ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.
ПК 1.2	ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.
ПК 1.3	ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.
ПК 1.4	ПК 1.4. Выполнять тестирование программных модулей.
ПК 1.5	ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода
ПК 2.4	ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.
ПК 2.5	ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

В результате освоения учебной дисциплины обучающийся осваивает элементы основных видов деятельности:

Основные виды деятельности	Код и формулировка компетенции	Показатели освоения компетенции
Разработка модулей программного обеспечения для компьютерных систем.	ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	Практический опыт: Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.
		Умения: Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства.

		<p><i>Дополнительно для квалификаций "Программист" и "Технический писатель":</i> Оценка сложности алгоритма.</p>
	<p>ПК 1.2. Разрабатывать программ-ные модули в соответствии с техни-ческим заданием.</p>	<p>Знания: Основные этапы разработки программ-ного обеспечения. Основные принципы технологии струк-турного и объектно-ориентированного программирования.</p> <p><i>Дополнительно для квалификаций "Про-граммист" и "Технический писатель":</i> Актуальная нормативно-правовая база в области документирования алгоритмов.</p> <p>Практический опыт: Разрабатывать код программного про-дукта на основе готовой спецификации на уровне модуля.</p> <p><i>Дополнительно для квалификаций "Про-граммист":</i> Разрабатывать мобильные приложения.</p> <p>Умения: Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программ-ные средства.</p> <p><i>Дополнительно для квалификаций "Про-граммист":</i> Осуществлять разработку кода про-граммного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.</p> <p><i>Дополнительно для квалификации "Спе-циалист по тестированию в области информационных технологий":</i> Осуществлять разработку модулей для различных видов тестирования.</p>

		<p>Знания: Основные этапы разработки программ-ного обеспечения. Основные принципы технологии струк-турного и объектно-ориентированного программирования.</p> <p><i>Дополнительно для квалификаций "Про-граммист":</i> Знание API современных мобильных операционных систем.</p>
--	--	--

<p>ПК.1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.</p>		<p>Практический опыт: Использовать инструментальные средства на этапе отладки программного продукта. Проводить тестирование программного модуля по определенному сценарию.</p> <p>Умения: Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификаций "Программист" и "Специалист по тестированию в области информационных технологий":</i> Применять инструментальные средства отладки программного обеспечения.</p> <p>Знания: Основные принципы отладки и тестирования программных продуктов. Инструментарий отладки программных продуктов.</p>
<p>ПК 1.4. Выполнять тестирование программных модулей.</p>		<p>Практический опыт: Проводить тестирование программного модуля по определенному сценарию. Использовать инструментальные средства на этапе тестирования программного продукта.</p> <p><i>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий":</i> Проводить тестирование в соответствии с функциональными требованиями.</p> <p>Умения: Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.</p> <p><i>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий":</i> Выполнять тестирование в соответствии с функциональными требованиями. Выполнять оценку тестового покрытия.</p>

		<p>Знания: Основные виды и принципы тестирования программных продуктов.</p> <p><i>Дополнительно для квалификации "Специалист по тестированию в области информационных технологий":</i> Методы организации работы при проведении функционального тестирования.</p>
--	--	--

	ПК 1.5. Осуществлять рефакто-ринг и оптимизацию программно-го кода.	Практический опыт: Анализировать алгоритмы, в том числе с применением инструментальных средств. Осуществлять рефакторинг и оптимизацию программного кода.
		Умения: Выполнять оптимизацию и рефакторинг программного кода. Работать с системой контроля версий.
		Знания: Способы оптимизации и приемы рефак-торинга. Инструментальные средства анализа ал-горитма. Методы организации рефакторинга и оп-тимизации кода. Принципы работы с системой контроляверсий.

	ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.	Практический опыт: Разрабатывать тестовые наборы (пакеты) для программного модуля. Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.
		Умения: Использовать выбранную систему контроля версий. Анализировать проектную и техниче-скую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системахконтроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тесто-вые сценарии Выполнять ручное.

		и автоматизированноетестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций.
		Знания: Модели процесса разработки программ-ного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированиюпрограммных модулей. Основы верификации и аттестации про-граммного обеспечения. Методы и способы идентификации сбоеви ошибок при интеграции приложений. Методы и схемы обработки исключи-тельных ситуаций. Основные методы и виды тестирования программных продуктов. Приемы работы с

		инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.
	ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	Практический опыт: Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.
		Умения: Использовать выбранную систему контроля версий. Использовать методы для получения кода с заданной функциональностью и степенью качества. Анализировать проектную и техническую документацию. Организовывать постобработку данных. Приемы работы в системах контроля версий. Выявлять ошибки в системных компонентах на основе спецификаций.
		Знания: Модели процесса разработки программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Стандарты качества программной документации.
		Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации работы в команде разработчиков.

Технологии формирования ОК

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

Предоставить обучающемуся возможность на практических занятиях для самостоятельного выбора способа решения задачи

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

Предоставить обучающемуся возможность на практических занятиях для самостоятельного выбора способа подготовки информации для решения задачи

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

Предоставить обучающемуся возможность на практических занятиях, на учебной и производственной практиках для решения задач в коллективе, помогать в случае затруднений

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

Предоставить обучающемуся возможность на лекциях практических занятиях для общения на государственном языке

ОК 09. Использовать информационные технологии в профессиональной деятельности.

Предоставить обучающимся возможности на практических занятиях выполнять задания средствами ИТ. Предоставлять студентам возможность самостоятельно осуществлять поиск, анализ и оценку информации при выполнении практической и самостоятельной работы

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке

Изучать профессиональные стандарты на государственном языке (русском) и знакомить обучающихся с международными стандартами на английском языке

Технологии формирования ПК

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием

Предоставить обучающимся возможность строить алгоритмы всех возможных видов. Научить определять вид алгоритма по ТЗ. Научить строить алгоритмы по ТЗ

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием

Предоставить обучающимся возможность научиться на примерах преобразовывать алгоритмы в программные модули. В дальнейшем, перейти к разработке программных модулей в соответствии с ТЗ.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств

Показать обучающимся возможности сред программирования для поиска синтаксических ошибок, возможности для прерываний с выводом значений переменных.

ПК 1.4. Выполнять тестирование программных модулей

Ознакомить обучающихся с видами и способами тестирования. Предоставить им возможность испытаний программных модулей в соответствии с разработанными этапами тестирования

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода

Продемонстрировать обучающимся примеры малопонятного и сложного кода, провести рефакторинг и оптимизацию его. Дать возможность выполнить такую работу самим обучающимся

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

Познакомить обучающихся с понятием тестового сценария и способами его реализации. Предоставить им возможность создать тестовые сценарии и опробовать их эффективность на практике

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

Ознакомить обучающихся с вариантами стандартов кодирования на разных языках программирования, описать историю развития стандартов, их трансформации. Требовать от обучающихся создания простого, ясного и эффективного программного кода.

2 СТРУКТУРА РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

1.1. Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем в часах
Объем образовательной программы	164
в том числе:	
теоретическое обучение	76
практические занятия	40
лабораторные занятия	36
консультации	4
<i>Самостоятельная работа</i>	2
Промежуточная аттестация (экзамен)	6

Максимальная учебная нагрузка обучающегося - 164 часа, в том числе

- обязательная аудиторная нагрузка - 153 часа;
- самостоятельная работа обучающегося - 2 часа;

2.2. Структура дисциплины

Наименование разделов и тем	Всего	Количество аудиторных часов		Самостоятельная работа обучающегося (час)
		Теоретическое обучение	Практические и лабораторные занятия	
Раздел 1. Введение в программирование	10	6	4	
Тема 1.1. Языки программирования	6	4	2	
Тема 1.2. Типы данных	4	2	2	
Раздел 2. Операторы языка программирования	36	14	22	
Тема 2.1. Операторы языка программирования	36	14	22	
Раздел 3. Усложнение программирования	22	10	12	
Тема 3.1. Процедуры и функции	10	4	6	
Тема 3.2. Структуризация в программировании	4	2	2	
Тема 3.3. Модульное программирование	8	4	4	
Раздел 4. Основные конструкции языков программирования	6	4	2	

Тема 4.1. Указатели	6	4	2	
Раздел 5. Применение ООП	78	42	36	
Тема 5.1. Основные принципы объектно-ориентированного программирования	8	8		
Тема 5.2. Интегрированная среда разработчика	14	12	2	
Тема 5.3. Визуальное событийно-ориентированное программирование	18	6	12	
Тема 5.4. Разработка оконного приложения	14	6	8	
Тема 5.5. Этапы разработки приложения	8	4	4	
Тема 5.6. Иерархия классов	16	6	10	
Всего по дисциплине				

2.3. Тематический план и содержание учебной дисциплины ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Тематический план и содержание учебной дисциплины «ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

Наименование разделов и тем		Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающегося, курсовая работа (проект)	Объем в часах		
1		2	3	4	
Раздел 1		Введение в программирование	10		
Тема 1.1. Языки программирования		Содержание учебного материала	6		
	Л1	1. Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы	2		
	Л2	4. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере	2		
		В том числе практических занятий и лабораторных работ			
		<i>Знакомство со средой программирования.</i>	2		Лаб 1
		Самостоятельная работа обучающихся			
Тема 1.2. Типы данных		Содержание учебного материала	4		
	Л3	1. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных	2		
		В том числе практических занятий и лабораторных работ			
		2		Лаб 2
		Самостоятельная работа обучающихся			
Раздел 2.		Содержание учебного материала	36		
Тема 2.1. Операторы языка программирования			36		
	Л 4	1. Операции и выражения. Правила формирования и вычисления выражений.	2		
	Л 5	Структура программы. Ввод и вывод данных	2		
	Л 6	3. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Массивы. Двумерные массивы	2		
	Л 7	4. Строки. Стандартные процедуры и функции для работы со строками.	2		
	Л8	5. Структурированный тип данных – множество. Операции над множествами.	2		
	Л9	6. Комбинированный тип данных – запись.	2		

	Л 10	Файлы последовательного доступа. Файлы прямого доступа	2		
		В том числе практических занятий и лабораторных работ			
		Математические и логические выражения	2		Лаб 3
		<i>Составление программ линейной структуры</i>	2	Пр 1	
		<i>Составление программ разветвляющейся структуры</i>	2	Пр 2	
		<i>Составление программ циклической структуры</i>	2	Пр 3	
		<i>Обработка одномерных массивов</i>	2		Лаб 4
		<i>Обработка двумерных массивов</i>	2		Лаб 5
		<i>Работа со строками</i>	2		Лаб 6
		<i>Работа с данными типа множество</i>	2		Лаб 7
		<i>Файлы последовательного доступа</i>	2	Пр 4	
		<i>Типизированные файлы</i>	2		Лаб 8
		<i>Нетипизированные файлы</i>	2		Лаб 9
		Самостоятельная работа обучающихся			
Раздел 3.		Подпрограммы	22		
Тема 3.1. Процедуры и функции		Содержание учебного материала	10		
	Л 11	1. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация процедур и функций.	2		
	Л 12	2. Рекурсия. Программирование рекурсивных алгоритмов.	2		
		В том числе практических занятий и лабораторных работ			
		<i>Организация процедур</i>	2		Лаб 10
		<i>Организация функций</i>	2		Лаб 11
		<i>Применение рекурсивных функций</i>	2	Пр 5	
		Самостоятельная работа обучающихся			
Тема 3.2. Структуризация в программировании		Содержание учебного материала	4		
	Л 13	1. Основы структурного программирования. Методы структурного программирования.	2		
		В том числе практических занятий и лабораторных работ			
		<i>Реализация структурных алгоритмов</i>	2	Пр 6	
		Самостоятельная работа обучающихся			
Тема 3.3.		Содержание учебного материала	8		

Модульное программирование					
	Л 14	1. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы.	2		
	Л 15	Стандартные модули.	2		
		В том числе практических занятий и лабораторных работ			
		<i>Программирование модуля</i>	2	Пр 7	
		<i>Создание библиотеки подпрограмм</i>	2	Пр 8	
		Самостоятельная работа обучающихся			
Раздел 4		Основные конструкции языков программирования	6		
Тема 4.1. Указатели		Содержание учебного материала	6		
	Л 16	1. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	2		
	Л 17	2. Структуры данных на основе указателей. Задача о стеке.	2		
		В том числе практических занятий и лабораторных работ			
		<i>Создание библиотеки подпрограмм</i>	2		Лаб 12
		Самостоятельная работа обучающихся			
Раздел 5			78		
Тема 5.1 Основные принципы объектно-ориентированного программирования(ООП)		Содержание учебного материала	8		
	Л 18	1. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс.	2		
	Л 19	2. Основные принципы ООП: инкапсуляция, наследование, полиморфизм.	2		
	Л 20	3. Классы объектов. Компоненты и их свойства.	2		
	Л 21	4. Событийно-управляемая модель программирования. Компонентно-ориентированный подход.	2		
		В том числе практических занятий и лабораторных работ			
		Самостоятельная работа обучающихся			
Тема 5.2 Интегрированная среда разработчика		Содержание учебного материала	14		
	Л 22	1. Требования к аппаратным и программным средствам интегрированной среды разработчика.	2		

	Л 23	2. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов.	2		
	Л 24	3. Панель компонентов и их свойства. Окно кода проекта.	2		
	Л 25	4. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	2		
	Л 26	5. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта.	2		
	Л 27	6. Настройка среды и параметров проекта.	2		
		В том числе практических занятий и лабораторных работ			
		<i>Изучение интегрированной среды разработчика</i>	2	Пр 9	
		Самостоятельная работа обучающихся			
Тема 5.3. Визуальное событийно-управляемое программирование		Содержание учебного материала	18		
	Л 28	1. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение.	2		
	Л 29	2. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства.	2		
	Л 30	3. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	2		
		В том числе практических занятий и лабораторных работ			
		Создание проекта с использованием компонентов для работы с текстом.	2		Лаб 13
		Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.	2		Лаб 14
		События компонентов (элементов управления), их сущность и назначение.	2		Лаб 15
		Создание процедур на основе событий.	2	Пр 10	
		Создание проекта с использованием кнопочных компонентов.	2	Пр 11	
		Создание проекта с использованием компонентов стандартных диалогов и системы меню	2	Пр 12	
		Самостоятельная работа обучающихся			
Тема 5.4 Разработка оконного приложения		Содержание учебного материала	14		
	Л 31	1. Разработка функционального интерфейса приложения. Создание интерфейса приложения.	2		

	Л 32	2. Разработка функциональной схемы работы приложения.	2		
	Л 33	3. Разработка игрового приложения.	2		
		В том числе практических занятий и лабораторных работ			
		Разработка функциональной схемы работы приложения.	2	Пр 13	
		Разработка оконного приложения с несколькими формами.	2	Пр 14	
		Разработка игрового приложения.	2	Пр 15	
		Создание процедур обработки событий. Компиляция и запуск приложения	2	Пр 16	
		Самостоятельная работа обучающихся			
Тема 5.5 Этапы разработки приложений		Содержание учебного материала	8		
	Л 34	1. Разработка приложения.	2		
	Л 35	2. Проектирование объектно-ориентированного приложения.	2		
		Разработка интерфейса приложения. Тестирование, отладка приложения	2	Пр 17	
		В том числе практических занятий и лабораторных работ			
		Создание интерфейса пользователя. Отладка и тестирование приложения	2	Пр 18	
Тема 5.6 Иерархия классов		Содержание учебного материала	16		
	Л 36	1. Классы ООП: виды, назначение, свойства, методы, события.	2		
	Л 37	2. Перегрузка методов.	2		
	Л 38	2		
		В том числе практических занятий и лабораторных работ			
		Классы ООП: виды, назначение, свойства, методы, события.	2		Лаб 16
		Объявления класса.	2		Лаб 17
		Создание наследованного класса.	2		Лаб 18
		Программирование приложений.	2	Пр 19	
		Перегрузка методов	2	Пр 20	
		164 = 2с + 4кон+ 76лекц + 40 пр + 36 лаб + 6экз			
		38лекц + 20 пр + 18 лаб = 76 пар			

2.4. Содержание разделов дисциплины

2.4.1. Занятия лекционного типа

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Введение в программирование	<p>Тема 1.1. Языки программирования</p> <p>Лекция 1. Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы</p> <p>Лекция 2. Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере</p> <p>Тема 1.2. Типы данных</p> <p>Лекция 3. Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных</p>	Т, У
2	Реализация простых алгоритмов средствами ЯП Паскаль	<p>Тема 2.1. Операторы языка программирования</p> <p>Лекция 4. Операции и выражения. Правила формирования и вычисления выражений</p> <p>Лекция 5. Структура программы. Ввод и вывод данных</p> <p>Лекция 6. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Массивы. Двумерные массивы</p> <p>Лекция 7. Строки. Стандартные процедуры и функции для работы со строками</p> <p>Лекция 7. Строки. Стандартные процедуры и функции для работы со строками</p> <p>Лекция 9. Комбинированный тип данных – запись</p> <p>Лекция 10. Файлы последовательного доступа. Файлы прямого доступа</p>	Т, У
3	Подпрограммы	<p>Тема 3.1. Процедуры и функции</p> <p>Лекция 11. Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация процедур и функций</p> <p>Лекция 12. Рекурсия. Программирование рекурсивных алгоритмов</p> <p>Тема 3.2. Структуризация программ</p> <p>Лекция 13. Основы структурного программирования. Методы структурного программирования</p> <p>Тема 3.3. Лекция 14. Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы</p> <p>Лекция 15. Стандартные модули</p>	Т, У
4	Основные конструкции и языков программирования	<p>Тема 4.1. Указатели</p> <p>Лекция 16. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных</p> <p>Лекция 16. Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных</p>	Т, У
5	Основы объектно-ориентированного программирования	<p>Тема 5.1. Основные принципы ООП</p> <p>Лекция 18. История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс</p> <p>Лекция 19. Основные принципы ООП: инкапсуляция, наследование, полиморфизм</p> <p>Лекция 20. Классы объектов. Компоненты и их свойства</p> <p>Лекция 21. Событийно-управляемая модель программирования.</p>	Т, У

	<p>Компонентно-ориентированный подход</p> <p>Тема 5.2. Интегрированная среда разработчика</p> <p>Лекция 22. Требования к аппаратным и программным средствам интегрированной среды разработчика</p> <p>Лекция 23. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов</p> <p>Лекция 24. Панель компонентов и их свойства. Окно кода проекта</p> <p>Лекция 25. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта</p> <p>Лекция 26. Панель компонентов и их свойства. Окно кода проекта. Состав и характеристика проекта. Выполнение проекта. Настройка среды и параметров проекта</p> <p>Лекция 27. Настройка среды и параметров проекта</p> <p>Тема 5.3. Визуальное событийно-управляемое программирование</p> <p>Лекция 28. Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение</p> <p>Лекция 29. Дополнительные элементы управления. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства</p> <p>Лекция 30. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий</p> <p>Тема 5.4. Разработка оконного приложения</p> <p>Лекция 31. Разработка функционального интерфейса приложения. Создание интерфейса приложения</p> <p>Лекция 32. Разработка функциональной схемы работы приложения</p> <p>Лекция 33. Разработка игрового приложения</p> <p>Тема 5.5. Этапы разработки приложений</p> <p>Лекция 34. Разработка приложения</p> <p>Лекция 35. Проектирование объектно-ориентированного приложения</p> <p>Тема 5.6. Иерархия классов</p> <p>Лекция 36. Классы ООП: виды, назначение, свойства, методы, события</p> <p>Лекция 37. Перегрузка методов</p> <p>Лекция 38. Наследование в классах</p>	
--	--	--

2.4.2. Занятия семинарского типа

Не предусмотрено

2.4.3. Практические занятия

№	Наименование раздела	Наименование практических (лабораторных) работ	Форма текущего контроля
<i>6 семестр</i>			
1	2	3	4
1.	Введение в программирование	<i>Лабораторная работа 1. Знакомство со средой программирования</i>	ПР, У
		<i>Лабораторная работа 2. Определение типов данных</i>	
2.	Операторы языка программирования	<i>Лабораторная работа 3. Математические и логические выражения</i>	ПР, У

		<i>Составление программ линейной структуры</i>	ПР, У
		<i>Составление программ разветвляющейся структуры</i>	ПР, У
		<i>Составление программ циклической структуры</i>	ПР, У
		<i>Лабораторная работа 4. Обработка одномерных массивов</i>	
		<i>Лабораторная работа 5. Обработка двумерных массивов</i>	ПР, У
		<i>Лабораторная работа 6. Работа со строками</i>	ПР, У
		<i>Лабораторная работа 7. Работа с данными типа множество</i>	ПР, У
		<i>Практическая работа 4. Файлы последовательного доступа</i>	ПР, У
		<i>Лабораторная работа 8. Типизированные файлы</i>	ПР, У
		<i>Лабораторная работа 9. Нетипизированные файлы</i>	ПР, У
3	Подпрограммы	<i>Лабораторная работа 10. Организация процедур</i>	ПР, У
		<i>Лабораторная работа 11. Организация функций</i>	ПР, У
		<i>Практическая работа 5. Применение рекурсивных функций</i>	ПР, У
		<i>Практическая работа 6. Реализация структурных алгоритмов</i>	ПР, У
		<i>Практическая работа 7. Программирование модуля</i>	ПР, У
		<i>Практическая работа 8. Создание библиотеки подпрограмм</i>	ПР, У
4	Указатели	<i>Лабораторная работа 12. Создание библиотеки подпрограмм</i>	ПР, У
5	Основы ООП	<i>Практическая работа 9. Изучение интегрированной среды разработчика</i>	ПР, У
		<i>Лабораторная работа 13. Создание проекта с использованием компонентов для работы с текстом.</i>	ПР, У
		<i>Лабораторная работа 14. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.</i>	ПР, У
		<i>Лабораторная работа 15. События компонентов (элементов управления), их сущность и назначение.</i>	ПР, У
		<i>Практическая работа 10.. Создание процедур на основе событий.</i>	ПР, У
		<i>Практическая работа 11. Создание проекта с использованием кнопочных компонентов.</i>	ПР, У
		<i>Практическая работа 12. Создание проекта с использованием компонентов стандартных диалогов и системы меню</i>	ПР, У
		<i>Практическая работа 13. Разработка функциональной схемы работы приложения.</i>	ПР, У
		<i>Практическая работа 14. Разработка оконного приложения с несколькими формами.</i>	ПР, У
		<i>Практическая работа 15. Разработка игрового приложения.</i>	ПР, У
		<i>Практическая работа 16. Создание процедур обработки событий. Компиляция и запуск приложения</i>	ПР, У
		<i>Практическая работа 17. Разработка интерфейса приложения. Тестирование, отладка приложения</i>	ПР, У
		<i>Практическая работа 18. Создание интерфейса пользователя. Отладка и тестирование приложения</i>	ПР, У
		<i>Лабораторная работа 16. Классы ООП: виды, назначение, свойства, методы, события.</i>	ПР, У
	<i>Лабораторная работа 17. Объявления класса.</i>	ПР, У	

		Лабораторная работа 18. Создание наследованного класса.	ПР, У
		<i>Практическая работа 19.</i> Программирование приложений.	ПР, У
		<i>Практическая работа 20.</i> Перегрузка методов	ПР, У
		Итоговая аттестация: экзамен	

2.4.4. Содержание самостоятельной работы

Подготовка к экзамену

Итоговый тест

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

3.1. Образовательные технологии при проведении лекций

Для реализации компетентного подхода предусматривается использование в учебном процессе активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

Преподавание дисциплины организовано по модульно-блочному принципу.

В процессе преподавания применяются образовательные технологии развития абстрактного, логического и критического мышления. Обязательны компьютерные практикумы по разделам дисциплины, тестирование, тематические презентации, интерактивные технологии.

В учебном процессе наряду с традиционными образовательными технологиями используются компьютерное

№	Тема	Виды применяемых образовательных технологий	Кол-во час
1	2	3	4
1	Введение в программирование	МБТ, ИКТ	10*
2	Операторы языка программирования	МБТ, ИКТ	36*
3	Подпрограммы	МБТ, ИКТ	22*
4	Основные конструкции языков программирования		6*
5	Основы ООП		78*
Итого по курсу			152*
в том числе интерактивное обучение при необходимости*			

3.2. Образовательные технологии при проведении практических занятий (лабораторных работ)

№	Тема занятия	Виды технологий	Часы
1	Лабораторная работа 1. Знакомство со средой программирования	ИКТ	2
2	Лабораторная работа 2. Определение типов данных	ИКТ	2
3	Лабораторная работа 3. Математические и логические выражения	ИКТ	2
4	Составление программ линейной структуры	ИКТ, работа в группах	2
5	Составление программ разветвляющейся структуры	ИКТ, работа в группах	2
6	Составление программ циклической структуры	ИКТ, работа в группах	2
7	Лабораторная работа 4. Обработка одномерных массивов	ИКТ	2
8	Лабораторная работа 5. Обработка двумерных массивов	ИКТ	2
9	Лабораторная работа 6. Работа со строками	ИКТ	2
10	Лабораторная работа 7. Работа с данными типа множество	ИКТ	2
11	Практическая работа 4. Файлы последовательного доступа	ИКТ, работа в	2

		группах	
12	Лабораторная работа 8. Типизированные файлы	ИКТ	2
13	Лабораторная работа 9. Нетипизированные файлы	ИКТ	2
14	Лабораторная работа 10. Организация процедур	ИКТ	2
15	Лабораторная работа 11. Организация функций	ИКТ	2
16	Практическая работа 5. Применение рекурсивных функций	ИКТ, работа в группах	2
17	Практическая работа 6. Реализация структурных алгоритмов	ИКТ, работа в группах	2
18	Практическая работа 7. Программирование модуля	ИКТ, работа в группах	2
19	Практическая работа 8. Создание библиотеки подпрограмм	ИКТ, работа в группах	2
20	Лабораторная работа 12. Создание библиотеки подпрограмм	ИКТ	2
21	Практическая работа 9. Изучение интегрированной среды разработчика	ИКТ, работа в группах	2
22	Лабораторная работа 13. Создание проекта с использованием компонентов для работы с текстом.	ИКТ	2
23	Лабораторная работа 14. Создание проекта с использованием компонентов ввода и отображения чисел, дат и времени.	ИКТ	2
24	Лабораторная работа 15. События компонентов (элементов управления), их сущность и назначение.	ИКТ	2
25	Практическая работа 10.. Создание процедур на основе событий.	ИКТ, работа в группах	2
26	Практическая работа 11. Создание проекта с использованием кнопочных компонентов.	ИКТ, работа в группах	2
27	Практическая работа 12. Создание проекта с использованием компонентов стандартных диалогов и системы меню	ИКТ, работа в группах	2
28	Практическая работа 13. Разработка функциональной схемы работы приложения.	ИКТ, работа в группах	2
29	Практическая работа 14. Разработка оконного приложения с несколькими формами.	ИКТ, работа в группах	2
30	Практическая работа 15. Разработка игрового приложения.	ИКТ, работа в группах	2
31	Практическая работа 16. Создание процедур обработки событий. Компиляция и запуск приложения	ИКТ, работа в группах	2
32	Практическая работа 17. Разработка интерфейса приложения. Тестирование, отладка приложения	ИКТ, работа в группах	2
33	Практическая работа 18. Создание интерфейса пользователя. Отладка и тестирование приложения	ИКТ, работа в группах	2
34	Лабораторная работа 16. Классы ООП: виды, назначение, свойства, методы, события.	ИКТ	2
35	Лабораторная работа 17. Объявления класса.	ИКТ	2
36	Лабораторная работа 18. Создание наследованного класса.	ИКТ	2
37	Практическая работа 19. Программирование приложений.	ИКТ, работа в группах	2
38	Практическая работа 20. Перегрузка методов	ИКТ, работа в группах	2
	Всего по курсу		76
	из них интерактивных (по необходимости)		76

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП.04. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения: Лаборатория программирования и баз данных

Оснащение лаборатории:

- мультимедийный проектор,
- экран;
- компьютеры обучающихся;
- компьютер преподавателя;
- сервер;
- локальная сеть с доступом в Интернет;
- программное обеспечение общего и профессионального назначения; учебно-методические материалы по дисциплине,
- электронные образовательные ресурсы по дисциплине,
- наглядные пособия по дисциплине;
- учебная мебель;
- маркерная доска;
- кондиционер;
- жалюзи.

4.2. Информационное обеспечение реализации программы

Для реализации программы библиотечный фонд образовательной организации должен иметь печатные и/или электронные образовательные и информационные ресурсы, рекомендуемых для использования в образовательном процессе

5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ, НЕОБХОДИМОЙ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

5.1 Основная литература

1. Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1150328>. – Режим доступа: по подписке.
2. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1735805>. – Режим доступа: по подписке.
3. Семакин, И. Г. Основы алгоритмизации и программирования : учебник для студентов учреждений среднего профессионального образования / И. Г. Семакин, А. П. Шестаков. - 5-е изд., стер. - Москва : Академия, 2021. - 301 с. - (Профессиональное образование. ТОП-50). - Библиогр.: с. 298-299. - ISBN 978-5-4468-9989-0. - Текст : непосредственный.

5.2 Дополнительная литература

1. Гуриков, С. Р. Основы алгоритмизации и программирования на языке Microsoft Visual Basic : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 594 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014442-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1864235>. – Режим доступа: по подписке.
2. Демин, А. Ю. Информатика. Лабораторный практикум : учебное пособие для среднего профессионального образования / А. Ю. Демин, В. А. Дорофеев. — Москва : Издательство Юрайт, 2023. — 133 с. — (Профессиональное образование). — ISBN 978-5-534-07984-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/516857>
3. Казанский, А. А. Объектно-ориентированный анализ и программирование на Visual Basic 2013 : учебник для среднего профессионального образования / А. А. Казанский. — Москва : Издательство Юрайт, 2023. — 290 с. — (Профессиональное образование). — ISBN 978-5-534-03833-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513399>
4. Фризен, И. Г. Основы алгоритмизации и программирования (среда PascalABC.NET) : учебное пособие / И.Г. Фризен. — Москва : ФОРУМ : ИНФРА-М, 2023. — 392 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-005-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1902735>. – Режим доступа: по подписке.
5. Макарова, Н. В., Основы программирования : учебник и практикум / Н. В. Макарова, Ю. Н. Нилова, С. Б. Зеленина, Е. В. Лебедева. — Москва : КноРус, 2021. — 452 с. — ISBN 978-5-406-03394-4. — URL: <https://book.ru/book/936582>. — Текст : электронный.
6. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2023. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/513113>

5.3 Периодические издания

1. Открытые системы.- URL: <http://biblioclub.ru/index.php?page=journal&jid=436083>
2. Информатика в школе .- URL: <http://dlib.eastview.com/browse/publication/18988>
3. Программные продукты и системы.- URL: <http://dlib.eastview.com/browse/publication/64086>
4. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
5. Системный администратор.- URL: <http://dlib.eastview.com/browse/publication/66751>
6. ComputerwordРоссия.- URL: <http://dlib.eastview.com/browse/publication/64081>

7. Мир ПК.- URL: <http://dlib.eastview.com/browse/publication/64067>
8. Информационно-управляющие системы.- URL: <http://dlib.eastview.com/browse/publication/71235>
9. Журнал сетевых решений LAN.- URL: <http://dlib.eastview.com/browse/publication/64078>
10. Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
11. Windows IT Pro/ Re.- URL: <http://biblioclub.ru/index.php?page=journal&jid=138741>

5.4 Перечень ресурсов информационно-телекоммуникационной сети Интернет

1. ЭБС «Университетская библиотека ONLINE»: сайт. – URL:<http://biblioclub.ru>
2. ЭБС Издательства «Лань»: сайт. – URL:<http://e.lanbook.com>
3. ЭБС «Юрайт»: сайт. –URL:<https://urait.ru/>
4. ЭБС «BOOK.ru»: сайт. – URL: <https://www.book.ru>
5. ЭБС «ZNANIUM.COM»: сайт. – URL: <https://www.znanium.com>
6. Базы данных компании «Ист Вью»: сайт. –URL: <http://dlib.eastview.com>
7. Научная электронная библиотека «eLibrary.ru»: сайт. – URL: <http://elibrary.ru/>
8. Электронная библиотека "Издательского дома "Гребенников". - URL: <http://www.grebennikon.ru/>
9. Университетская информационная система РОССИЯ (УИС Россия). - URL: <http://uisrussia.msu.ru/>
10. "Лекториум ТВ" - видеолекции ведущих лекторов России. - URL: <http://www.lektorium.tv/>
11. База учебных планов, учебно-методических комплексов, публикаций и конференций КубГУ. - URL: <http://docspace.kubsu.ru/>
12. Российское образование [Федеральный портал]. - URL: <https://www.edu.ru/>

6. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

«ОП.04 Основы алгоритмизации и программирования»

6.1. Критерии оценивания практических работ

<i>Результаты обучения</i>	<i>Критерии оценки</i>	<i>Формы и методы контроля</i>
<ul style="list-style-type: none"> – Правовые основы стандартизации – Основные понятия и определения стандартизации,. – Основные положения систем (комплексов) общетехнических и организационно-методических стандартов. – Показатели качества и методы их оценки – Системы качества – Основные термины и определения в области сертификации. – Организационную структуру сертификации. – Системы и схемы сертификации. <p>Перечень умений, осваиваемых в рамках дисциплины:</p> <ul style="list-style-type: none"> – Применять требования нормативных актов к основным видам продукции (услуг) и процессов. – Применять документацию систем качества. – Применять основные правила и документы системы сертификации Российской Федерации. 	<p>«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко.</p> <p>«Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками.</p> <p>«Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным материалом в основном сформированы, большинство предусмотр. программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.</p> <p>«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>Примеры форм и методов контроля и оценки</p> <ul style="list-style-type: none"> – Компьютерное тестирование на знание терминологии по теме; – Тестирование.... – Контрольная работа – Самостоятельная работа. – Защита реферата.... – Семинар – Защита курсовой работы (проекта) – Выполнение проекта; – Наблюдение за выполнением практического задания. (деятельностью студента) – Оценка выполнения практического задания(работы) – Подготовка и выступление с докладом, сообщением, презентацией... – Решение ситуационной задачи....

Отметка «5» ставится, если:

- работа выполнена полностью, в логических рассуждениях и обосновании решения нет пробелов и ошибок;
- в решении нет ошибок (возможна одна неточность, описка, не являющаяся следствием незнания или непонимания учебного материала).

Отметка «4» ставится, если:

- работа выполнена полностью, но в обосновании шагов решения недостаточны;
- допущена 1-2 ошибки или 1 ошибка и два-три недочета в выкладках.

Отметка «3» ставится, если:

- допущены 3 ошибки или 2 ошибки и более двух-трех недочетов в выкладках, но обучающийся владеет обязательными умениями по проверяемой теме.

Отметка «2» ставится, если:

- допущены существенные ошибки, показавшие, что обучающийся не владеет обязательными умениями по данной теме в полной мере

6.2. Критерии оценивания конспектов

Отметка «5» ставится, если:

- работа содержит полные ответы на все теоретические вопросы для составления конспекта;

Отметка «4» ставится, если:

- работа содержит неполный ответ хотя бы на один теоретический вопрос для составления конспекта;

Отметка «3» ставится, если:

- работа содержит неполные ответы на 2 теоретических вопроса для составления конспекта.

Отметка «2» ставится, если:

- работа содержит неполные ответы на 2 и более теоретических вопроса для составления конспекта.

6.3. Критерии оценивания презентаций

Оценка	5	4	3	2
Содержание	Работа полностью завершена	Почти полностью сделаны наиболее важные компоненты работы	Не все важнейшие компоненты работы выполнены	Работа сделана фрагментарно и с помощью преподавателя
	Работа демонстрирует глубокое	Работа демонстрирует понимание	Работа демонстрирует понимание, но	Работа демонстрирует минимальное

	понимание описываемых процессов	основных моментов, хотя некоторые детали не уточняются	неполное	понимание
	Грамотно используется научная лексика	Научная лексика используется, но иногда не корректно	Научная терминология или используется мало или используется некорректно.	Минимум научных терминов
Грамотность	Нет ошибок: ни грамматических, ни синтаксических	Минимальное количество ошибок	Есть ошибки, мешающие восприятию	Много ошибок, делающих материал трудночитаемым
Дизайн	Имеются постоянные элементы дизайна. Дизайн подчеркивает содержание	Имеются постоянные элементы дизайна. Дизайн соответствует содержанию.	Нет постоянных элементов дизайна. Дизайн может и не соответствовать содержанию.	Элементы дизайна мешают содержанию, накладываясь на него.
	Все параметры шрифта хорошо подобраны (текст хорошо читается)	Параметры шрифта подобраны. Шрифт читаем	Параметры шрифта недостаточно хорошо подобраны, могут мешать восприятию	Параметры не подобраны. Делают текст трудночитаемым

6.4 МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ МОДУЛЯ

Обучающиеся для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку, как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа является важнейшей формой учебно-познавательного процесса. Цель заданий для самостоятельной работы – закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Обучающийся должен изучить список нормативно-правовых актов и экономической литературы, рекомендуемый по учебной дисциплине; уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания обучающийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Обучение осуществляется по модульно-блочной технологии (лекции, практики) с включением инновационных элементов.

С точки зрения используемых методов лекции подразделяются следующим образом: информационно-объяснительная лекция, повествовательная, лекция-беседа, проблемная лекция и т. д.

Устное изложение учебного материала на лекции должно конспектироваться. Слушать лекцию нужно уметь – поддерживать своё внимание, понять и запомнить

услышанное, уловить паузы. В процессе изложения преподавателем лекции студент должен выяснить все непонятные вопросы. Записывать содержание лекции нужно обязательно – записи помогают поддерживать внимание, способствуют пониманию и запоминанию услышанного, приводят знание в систему, служат опорой для перехода к более глубокому самостоятельному изучению предмета.

Методические рекомендации по конспектированию лекций:

- запись должна быть системной, представлять собой сокращённый вариант лекции преподавателя. Необходимо слушать, обдумывать и записывать одновременно;

- запись ведётся очень быстро, чётко, по возможности короткими выражениями;

- не прекращая слушать преподавателя, нужно записывать то, что необходимо усвоить.

Нельзя записывать сразу же высказанную мысль преподавателя, следует её понять и после этого кратко записать своими словами или словами преподавателя. Важно, чтобы в ней не был потерян основной смысл сказанного;

- имена, даты, названия, выводы, определения записываются точно;

- следует обратить внимание на оформление записи лекции. Для каждого предмета заводится общая тетрадь. Отличным от остального цвета следует выделять отдельные мысли и заголовки, сокращать отдельные слова и предложения, использовать условные знаки, буквы латинского и греческого алфавитов, а также некоторые приёмы стенографического сокращения слов.

Практические занятия по дисциплине ОП.04 «Основы алгоритмизации и программирования» проводятся в основном по схеме:

- устный опрос по теории в начале занятия (обсуждение теоретических проблемных вопросов по теме);

- работа в группах по разрешению различных ситуаций по теме занятия;

- решение практических задач индивидуально;

- подведение итогов занятия (или рефлексия);

- индивидуальные задания для подготовки к следующим практическим занятиям.

Цель практического (или лабораторного) занятия - научить студентов применять теоретические знания при решении практических задач на основе реальных данных.

На практических занятиях преобладают следующие методы:

- вербальные (преобладающим методом должно быть объяснение);

- практические (письменные задания, групповые задания и т. п.).

Важным для студента является умение рационально подбирать необходимую учебную литературу. Основными литературными источниками являются:

- библиотечные фонды филиала КубГУ в г. Геленджике;

- электронная библиотечная система «Университетская библиотека онлайн»;

– электронная библиотечная система Издательства «Лань».

Поиск книг в библиотеке необходимо начинать с изучения предметного каталога и создания списка книг, пособий, методических материалов по теме изучения.

Просмотр книги начинается с титульного листа, следующего после обложки. На нём обычно помещаются все основные данные, характеризующие книгу: название, автор, выходные данные, данные о переиздании и т.д. На обороте титульного листа даётся аннотация, в которой указывается тематика вопросов, освещённых в книге, определяется круг читателей, на который она рассчитана. Большое значение имеет предисловие книги, которое знакомит читателя с личностью автора, историей создания книги, раскрывает содержание.

Прочитав предисловие и получив общее представление о книге, следует обратиться к оглавлению. Оглавление книги знакомит обучаемого с содержанием и логической структурой книги, позволяет выбрать нужный материал для изучения. Год издания книги позволяет судить о новизне материала. В книге могут быть примечания, которые содержат различные дополнительные сведения. Они печатаются вне основного текста и разъясняют отдельные вопросы. Предметные и алфавитные указатели значительно облегчают повторение изложенного в книге материала. В конце книги может располагаться вспомогательный материал. К нему обычно относятся инструкции, приложения, схемы, ситуационные задачи, вопросы для самоконтроля и т.д.

Для лучшего представления и запоминания материала целесообразно вести записи и конспекты различного содержания, а именно:

- пометки, замечания, выделение главного;
- план, тезисы, выписки, цитаты;
- конспект, рабочая записка, реферат, доклад, лекция и т.д.

Читать учебник необходимо вдумчиво, внимательно, не пропуская текста, стараясь понять каждую фразу, одновременно разбирая примеры, схемы, таблицы, рисунки, приведённые в учебнике.

Одним из важнейших средств, способствующих закреплению знаний, является краткая запись прочитанного материала – составление конспекта. Конспект – это краткое связное изложение содержания темы, учебника или его части, без подробностей и второстепенных деталей. По своей структуре и последовательности конспект должен соответствовать плану учебника. Поэтому важно сначала составить план, а потом писать конспект в виде ответа на вопросы плана. Если учебник разделён на небольшие озаглавленные части, то заголовки можно рассматривать как пункты плана, а из текста каждой части следует записать те мысли, которые раскрывают смысл заголовка.

Требования к конспекту:

- краткость, сжатость, целесообразность каждого записываемого слова;
- содержательность записи - записываемые мысли следует формулировать кратко, но без ущерба для смысла. Объём конспекта, как правило, меньше изучаемого текста в 7-15 раз;
- конспект может быть, как простым, так и сложным по структуре – это зависит от содержания книги и цели её изучения.

Методические рекомендации по конспектированию:

- прежде чем начать составлять конспект, нужно ознакомиться с книгой, прочитать её сначала до конца, понять прочитанное;
- на обложке тетради записываются название конспектируемой книги и имя автора, составляется план конспектируемого текста;
- записи лучше делать при прочтении не одного-двух абзацев, а целого параграфа или главы;
- конспектирование ведётся не с целью иметь определённые записи, а для более полного овладения содержанием изучаемого текста, поэтому в записях отмечается и выделяется всё то новое, интересное и нужное, что особенно привлекло внимание;
- после того, как сделана запись содержания параграфа, главы, следует перечитать её, затем снова обращаться к тексту и проверить себя, правильно ли изложено содержание.

Техника конспектирования:

- конспектируя книгу большого объёма, запись следует вести в общей тетради;
- на каждой странице слева оставляют поля шириной 25-30 мм для записи коротких подзаголовков, кратких замечаний, вопросов;
- каждая страница тетради нумеруется;
- для повышения читаемости записи оставляют интервалы между строками, абзацами, новую мысль начинают с «красной» строки;
- при конспектировании широко используют различные сокращения и условные знаки, но не в ущерб смыслу записанного. Рекомендуется применять общеупотребительные сокращения, например: м.б. – может быть; гос. – государственный; д.б. – должно быть и т.д.
- не следует сокращать имена и названия, кроме очень часто повторяющихся;
- в конспекте не должно быть механического переписывания текста без продумывания его содержания и смыслового анализа.

Для написания реферата необходимо выбрать тему, согласовать ее с

преподавателем, подобрать несколько источников по теме, выполнить анализ источников по решению проблемы, обосновать свою точку зрения на решение проблемы.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

7.1. Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	Введение в программирование	ОК 1, ОК 2, ОК 4, ОК 5 ОК 9, ОК 10 ПК 1.1 - ПК 1.5, ПК 2.4,- 2.5	Тестирование, анализ выполнения практических работ
2.	Операторы языка программирования	ОК 1, ОК 2, ОК 4, ОК 5 ОК 9, ОК 10 ПК 1.1 - ПК 1.5, ПК 2.4,- 2.	Тестирование, анализ выполнения практических работ
3.	Подпрограммы	ОК 1, ОК 2, ОК 4, ОК 5 ОК 9, ОК 10 ПК 1.1 - ПК 1.5, ПК 2.4,- 2.	Тестирование, анализ выполнения практических работ
4.	Основные конструкции языков программирования	ОК 1, ОК 2, ОК 4, ОК 5 ОК 9, ОК 10 ПК 1.1 - ПК 1.5, ПК 2.4,- 2.	Тестирование, анализ выполнения практических работ
5.	Основы ООП	ОК 1, ОК 2, ОК 4, ОК 5 ОК 9, ОК 10 ПК 1.1 - ПК 1.5, ПК 2.4,- 2.	Тестирование, анализ выполнения практических работ

7.2. Критерии оценки знаний

Контроль и оценка результатов освоения модуля осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения студентами индивидуальных самостоятельных заданий.

Тест. Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тест оценивается по количеству правильных ответов (не менее 50%).

Критерии оценки знаний в целом по дисциплине:

«отлично» - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы модуля и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

«хорошо» - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по

образцу в стандартной ситуации;

«неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания учебной программы модуля, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

7.3. Оценочные средств для проведения текущей аттестации

Текущий контроль проводится в форме:

- устный опрос;
- тестирование по теоретическому материалу;
- практическая работа;

Форма аттестации	Знания	Умения	Практический опыт (владение)	Личные качества обучающегося	Примеры оценочных средств
Устный опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературными источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программными средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков логического анализа и синтеза при сопоставлении конкретных понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

Список вопросов теста для зачёта

7.4. Оценочные средства для проведения промежуточной аттестации

7.4.1. Оценочные средства текущей аттестации

Тест по всем темам Разделов 1-3

Задание #1

Вопрос:

Что выполняет оператор Writeln(a)?

Выберите один из 5 вариантов ответа:

- 1) Вывод значения переменной a
- 2) Вывод значения переменной x
- 3) Ввод переменной a
- 4) Вставка пустой строки
- 5) запись неверная

Задание #2

Вопрос:

Установите соответствие

Укажите соответствие для всех 5 вариантов ответа:

- 1) целые числа
- 2) вещественные числа
- 3) строка
- 4) символ
- 5) логический

__ integer
__ real
__ string
__ char
__ boolean

Задание #3

Вопрос:

Установите соответствие

Укажите соответствие для всех 6 вариантов ответа:

- 1) описание переменных
- 2) подключение библиотечных модулей
- 3) название программы
- 4) окончание программы
- 5) описание констант
- 6) нет такого служебного слова

__ var
__ uses
__ program
__ end.
__ const
__ vars

Задание #4

Вопрос:

Расположите по порядку

Укажите порядок следования всех 6 вариантов ответа:

__ End.


```
__ Uses crt;
__ var x : real;
__ Begin
__ x := 100/(25+0.25);
__ writeln('x=', x : 6 : 2);
```

Задание #5

Вопрос:

Установите соответствие

Укажите соответствие для всех 4 вариантов ответа:

- 1) $2 * (\exp(3 * \ln(x)) - 1)$
- 2) $\exp(3 * \ln(x)) - 1$
- 3) $\text{sqrt}(\exp(3 * \ln(x)) - 1)$
- 4) $2 * \exp(3 * \ln(x)) - 1$

$$2x^3 - 1$$

—

$$x^3 - 1$$

—

$$(x^3 - 1)^2$$

—

$$2(x^3 - 1)$$

—

Задание #6

Вопрос:

Верно ли, библиотечные модули надо подключать сразу после названия программы? (да, нет)

Запишите ответ:

Задание #7

Вопрос:

Верно ли, что блок описания переменных может в программе отсутствовать? (да, нет)

Запишите ответ:

Задание #8

Вопрос:

Какой оператор выполняет очистку экрана?

Выберите один из 4 вариантов ответа:

- 1) clrscr
- 2) clrscr
- 3) clrscr
- 4) crslcr

Задание #9

Вопрос:

Фрагмент программы:

$$a := f / h;$$

Какой тип данных необходимо выбрать для переменной a?

Выберите один из 5 вариантов ответа:

- 1) integer
- 2) real
- 3) byte
- 4) word
- 5) char

Задание #10

Вопрос:

Фрагмент программы:

$$c := (x \bmod 100) \operatorname{div} 10;$$

Какой тип данных можно выбрать для переменной c, если известно, что c - положительное трехзначное число?

Выберите несколько из 6 вариантов ответа:

- 1) byte
- 2) word
- 3) integer
- 4) real
- 5) longint
- 6) string

Задание #11

Вопрос:

Расположите по порядку этапы решения задачи

Укажите порядок следования всех 5 вариантов ответа:

- ___ описание переменных
- ___ ввод известных значений
- ___ вычисление неизвестного
- ___ вывод результата
- ___ название программы

Задание #12

Вопрос:

Как правильно подключить библиотеку crt?

Выберите один из 5 вариантов ответа:

- 1) uses crt;
- 2) uses : crt;
- 3) uses; crt;
- 4) crt uses;
- 5) user crt;

Задание #13

Вопрос:

Расположите по порядку этапы решения следующей задачи:

Известна площадь квадрата (S) со стороной a. Найти периметр (P).

Укажите порядок следования всех 4 вариантов ответа:

- Ввод S
- Вычисление значения стороны
- Вычисление P
- Вывод P

Задание #14

Вопрос:

Установите соответствие

Укажите соответствие для всех 7 вариантов ответа:

- 1) \sqrt{x}
- 2) $\cos x^2$
- 3) $|x|$
- 4) e^x
- 5) x^2
- 6) $\cos^2 x$
- 7) неверное выражение

- abs(x)
- sqr(x)
- sqrt(x)
- exp(x)
- cos(sqr(x))
- sqr(cos(x))
- cos(x)^2

Задание #15

Вопрос:

Фрагмент программы:

```
x := 25.26;  
Writeln(_____);
```

Как записать формат вывода, для переменной x: общее количество знаков - 6, количество знаков после запятой - 1

Примечание. Пробелы в ответе не ставить! Ответ вводить с использованием латинских букв!

Запишите ответ:

Задание #16

Вопрос:

После вычисления был получен результат d = 1.126

Какой использовался формат для вывода, если на экране отобразилось число 1.13 ?

Выберите один из 5 вариантов ответа:

- 1) `Writeln(d:4:2)`
- 2) `Readln(d:4:2)`
- 3) `Writeln(d:2:4)`
- 4) `Writeln(d:1:2)`
- 5) `Writeln(d:0:2)`

Задание #17

Вопрос:

Допустим, что $a_1=1$, $a_2=2$, $a_3=3$, $a_4=4$

Как будут выведены на экран значения при выполнении следующей строки:

```
Writeln(a1,a2,a3,a4);
```

Выберите один из 6 вариантов ответа:

- 1) 1234
- 2) 1 2 3 4
- 3) 1
- 2
- 3
- 4
- 4) $a_1=1$ $a_2=2$ $a_3=3$ $a_4=4$
- 5) a11a22a33a44
- 6) нет правильного ответа

Задание #18

Вопрос:

Допустим, что $d=25$, $c=26$

Как будут выведены на экран значения при выполнении следующей строки:

```
Writeln(' d= ',d,' c= ',c);
```

Выберите один из 6 вариантов ответа:

- 1) 25 26
- 2) $d= 25$ $c= 26$
- 3) $c= 26$ $d= 25$
- 4) $d= 26$ $c= 25$
- 5) $d=25c=26$
- 6) нет правильного ответа

Задание #19

Вопрос:

Как вводить значения при выполнении следующей строки?

```
Read(a,b,c,d);
```

Выберите один из 5 вариантов ответа:

- 1) нет правильного ответа
- 2) 5 8 74 6
- 3) 5
- 8
- 74
- 6
- 4) 58746

5) 5,8,74,6

Задание #20

Вопрос:

Для чего используется оператор Readln без параметров?

Выберите один из 5 вариантов ответа:

- 1) Для задержки экрана
- 2) Для вставки пустой строки
- 3) Для ввода значений
- 4) Для вывода результатов
- 5) не используется оператор Readln без параметров

Задание #21

Вопрос:

Для чего используется оператор Writeln без параметров?

Выберите один из 5 вариантов ответа:

- 1) Для задержки экрана
- 2) Для вставки пустой строки
- 3) Для ввода значений
- 4) Для вывода результатов
- 5) не используется оператор Readln без параметров

Задание #22

Вопрос:

Какой тип данных выбрать для работы со следующими значениями?

Укажите соответствие для всех 5 вариантов ответа:

- 1) byte
- 2) так записывать число в Pascale нельзя
- 3) real
- 4) integer
- 5) word

- ___ 122
- ___ 12,65
- ___ 12.56
- ___ 12587
- ___ -56

Задание #23

Вопрос:

Как записать выражение?

$$c = \sqrt{a^2 + b^2}$$

Выберите один из 5 вариантов ответа:

- 1) c := sqrt (sqr(a)+sqr(b))
- 2) c = sqrt (sqr(a)+sqr(b))
- 3) c := sqrt (sqr(a)+sqr(b))
- 4) c := sqr (sqrt(a)+sqrt(b))
- 5) c := sqrt (sqr(a))+sqr(b)

Задание #24

Вопрос:

Установите соответствие

Укажите соответствие для всех 6 вариантов ответа:

- 1) квадрат числа
- 2) квадратный корень
- 3) вывод
- 4) ввод
- 5) библиотечный модуль
- 6) модуль

sqr
 sqrt
 writeln
 readln
 crt
 abs

Задание #25

Вопрос:

Установите соответствие:

Укажите соответствие для всех 5 вариантов ответа:

- 1) остаток от деления
- 2) целая часть от деления
- 3) квадрат числа
- 4) квадратный корень числа
- 5) абсолютное значение

mod
 div
 sqr
 sqrt
 abs

Задание #26

Вопрос:

Установите соответствие между выражением и результатом:

Укажите соответствие для всех 5 вариантов ответа:

- 1) 41
- 2) 3
- 3) 12
- 4) 1
- 5) 0

123 div 10
 123 mod 10
 2541 mod 100
 2541 mod 10
 502 mod 2

Задание #27

Вопрос:

Выберите верную запись для вычисления следующего задания:

$$f = \begin{cases} -x^2 - 1, & \text{если } 5 > x > 0 \\ \sqrt{2x - 1}, & \text{если } x \geq 5 \\ \sqrt{|x - 5|}, & \text{в других случаях} \end{cases}$$

Выберите один из 5 вариантов ответа:

- 1) If $(x > 0)$ and $(x < 5)$ then $f := -\text{sqr}(x) - 1$ else if $x \geq 5$ then $f := \text{sqrt}(2 * x - 1)$ else $f := \text{sqrt}(\text{abs}(x - 5))$;
- 2) If $(x > 0)$ or $(x < 5)$ then $f := -\text{sqr}(x) - 1$ else if $x \geq 5$ then $f := \text{sqrt}(2 * x - 1)$ else $f := \text{sqrt}(\text{abs}(x - 5))$;
- 3) If $x > 0$ and $x < 5$ then $f := -\text{sqr}(x) - 1$; if $x \geq 5$ then $f := \text{sqrt}(2 * x - 1)$ else $f := \text{sqrt}(\text{abs}(x - 5))$;
- 4) If $(x > 0)$ and $(x < 5)$ then $f := -\text{sqr}(x) - 1$ then if $x \geq 5$ then $f := \text{sqrt}(2 * x - 1)$ else $f := \text{sqrt}(\text{abs}(x - 5))$;
- 5) If $(x > 0)$ and $(x < 5)$ then $f := -\text{sqr}(x) - 1$ else if $x \geq 5$ then $f := \text{sqrt}(2 * x - 1)$ else $f := \text{sqrt}(\text{abs}(x - 5))$;

Задание #28

Вопрос:

$a := 5; b := 10;$

If $a > b$ then $c := (a \bmod 2) * b$ else $c := (a \text{ div } 2) * b$;

Чему равен результат? c=?

Выберите один из 5 вариантов ответа:

- 1) 20
- 2) 10
- 3) 25
- 4) 0
- 5) 50

Задание #29

Вопрос:

$a := 12; b := 10;$

If $a - b > 0$ then $c := \text{sqr}(a) - b$ else $c := \text{sqr}(b) - a$;

Чему равен результат? c=?

Выберите один из 5 вариантов ответа:

- 1) 134
- 2) 144
- 3) 88
- 4) 4
- 5) 154

Задание #30

Вопрос:

$a := 12; b := 10; d := 20$

If $a > b$ then if $a > d$ then $c := a + b + d$ else

if $b > d$ then if $b > a$ then $c := a * b - d$ else $c := (2 * a - b) * d$

else writeln('нет решений');

Чему равен результат? c=?

Выберите один из 5 вариантов ответа:

- 1) нет решений
- 2) 42
- 3) 10
- 4) 280
- 5) 22

Задание #31

Вопрос:

a := random (120);

Из какого диапазона будет выбрано значение для переменной a?

Выберите один из 5 вариантов ответа:

- 1) 0..120
- 2) 0..119
- 3) 1..120
- 4) 10..130
- 5) 100..220

Задание #32

Вопрос:

a := -18 + random (26);

Из какого диапазона будет выбрано значение для переменной a?

Выберите один из 5 вариантов ответа:

- 1) -18..+7
- 2) -18..+8
- 3) -18..0
- 4) 0..26
- 5) 0..8

Задание #33

Вопрос:

Верно ли, что при использовании оператора IF оператор ELSE можно не использовать? (да, нет)

Запишите ответ:

Задание #34

Вопрос:

Верно ли, что при использовании оператора IF оператор THEN можно не использовать? (да, нет)

Запишите ответ:

Задание #35

Вопрос:

Дано 2 числа (A,B). Вычислить частное от деления первого числа на второе.

Выберите более корректное решение задания.

Выберите один из 5 вариантов ответа:

- 1) C := A/B;
- 2) If B=0 then writeln('на ноль делить нельзя') else C := A/B;
- 3) If B=0 then C := A/B else writeln('на ноль делить нельзя');
- 4) If B=0 then writeln('на ноль делить нельзя'); else C := A/B;
- 5) If A=0 then writeln('на ноль делить нельзя') else C := A/B;

Задание #36

Вопрос:

Дано 2 числа (D,F). Если первое число больше второго, то вычислить сумму чисел, если второе число больше первого, то вычислить их произведение. В противном случае найти разность между первым и вторым числом.

Выберите правильный вариант решения задачи.

Выберите один из 5 вариантов ответа:

- 1) If $D > F$ then $S := D + F$ else if $D < F$ then $S := D * F$ else $S := D - F$;
- 2) If $D = F$ then $S := D + F$ else if $D < F$ then $S := D * F$ else $S := D - F$;
- 3) If $D > F$ then $S := D + F$ else if $D < F$ then $S := D * F$ else $S := F - D$;
- 4) If $D > F$ then $S := D + F$; else if $D < F$ then $S := D * F$ else $S := D - F$;
- 5) If $D > F$ then $S := D + F$ else if $F > D$ then $S := D * F$; else $S := D - F$;

Задание #37

Вопрос:

Выберите те задачи, для решения которых можно использовать оператор CASE.

Выберите несколько из 6 вариантов ответа:

- 1) По номеру дня недели определить, как он называется: "пн", "вт", "ср"....
- 2) Найти корни квадратного уравнения
- 3) По номеру месяца определить время года
- 4) По номеру месяца определить, сколько дней в заданном месяце
- 5) Дано 5 чисел. Вычислить их сумму, произведение и среднее арифметическое
- 6) Дано N чисел. Определить, сколько из них отрицательных

Задание #38

Вопрос:

Дано число. Определить, принадлежит ли оно диапазону от 10 до 50.

Выберите правильный вариант решения задачи.

Выберите один из 5 вариантов ответа:

- 1) If $(x \geq 10)$ and $(x \leq 50)$ then writeln("Число в диапазоне от 10 до 50") else writeln("Число меньше 10 или больше 50");
- 2) If $(x \geq 10)$ or $(x \leq 50)$ then writeln("Число в диапазоне от 10 до 50") else writeln("Число меньше 10 или больше 50");
- 3) If $x \geq 10$ and $x \leq 50$ then writeln("Число в диапазоне от 10 до 50") else writeln("Число меньше 10 или больше 50");
- 4) If $(x \geq 10)$ and $(y \leq 50)$ then writeln("Число в диапазоне от 10 до 50") else writeln("Число меньше 10 или больше 50");
- 5) If $x \geq 10$ then if $x \leq 50$ then writeln("Число меньше 10 или больше 50") else writeln("Число в диапазоне от 10 до 50");

Задание #39

Вопрос:

Дано число. Определить делится ли число на 2? на 3?

Выбрать правильный вариант решения.

Выберите один из 5 вариантов ответа:

- 1) If $x \bmod 2 = 0$ then writeln("Число делится на 2") else writeln("Число не делится на 2");
If $x \bmod 3 = 0$ then writeln("Число делится на 3") else writeln("Число не делится на 3");
- 2) If $x \bmod 2 = 0$ then writeln("Число делится на 2") else writeln("Число не делится на 3");
If $x \bmod 3 = 0$ then writeln("Число делится на 3") else writeln("Число не делится на 2");
- 3) If $x \bmod 2 = 1$ then writeln("Число делится на 2") else writeln("Число не делится на 2");
If $x \bmod 3 = 1$ then writeln("Число делится на 3") else writeln("Число не делится на 3");
- 4) If $x \bmod 2 = 0$ then writeln("Число делится на 2"); else writeln("Число не делится на 2");
If $x \bmod 3 = 0$ then writeln("Число делится на 3"); else writeln("Число не делится на 3");
- 5) If $x \bmod 2 = 0$ then writeln("Число делится на 3") else writeln("Число не делится на 2");
If $x \bmod 3 = 0$ then writeln("Число делится на 2") else writeln("Число не делится на 3");

Задание #40

Вопрос:

$x := 123$;

$c := (x \bmod 10) * 100 + ((x \div 10) \bmod 10) * 10 + (x \div 100)$;

Чему будет равно значение переменной c после выполнения вычисления?

Выберите один из 5 вариантов ответа:

- 1) 123
- 2) 321
- 3) 231
- 4) 331
- 5) 122

Задание #41

Вопрос:

Дано 3 переменных типа longint A,B,C. Какие допустимы выражения для таких переменных?

Выберите несколько из 8 вариантов ответа:

- 1) $C := A + B;$
- 2) $C := B / A;$
- 3) $B := A * C;$
- 4) $A := \text{SQRT}(B * C);$
- 5) $C := A - B;$
- 6) $A := B \text{ MOD } C;$
- 7) $C := B \text{ DIV } A;$
- 8) $B := (A + C) - 2 * A;$

Задание #42

Вопрос:

Как правильно записать: переменная K принимает значение из диапазона от -10 до +27?

Выберите один из 6 вариантов ответа:

- 1) $K := \text{random}(-10) + \text{random}(27);$
- 2) $K := -10 + \text{random}(27);$
- 3) $K := -10 + \text{random}(38);$
- 4) $K := -10 + \text{random}(37);$
- 5) $K := \text{random}(38) + 10;$
- 6) $K := -10 - \text{random}(38);$

Задание #43

Вопрос:

$x := 3681;$
 $c := (x \text{ mod } 100) - (x \text{ div } 100);$

Чему равно значение переменной c?

Выберите один из 5 вариантов ответа:

- 1) 45
- 2) -45
- 3) 55
- 4) 2
- 5) 5

Задание #44

Вопрос:

$x := 3681;$
 $t := ((x \text{ div } 100) \text{ mod } 10) + ((x \text{ mod } 100) \text{ div } 10);$

Чему равно значение переменной t?

Выберите один из 5 вариантов ответа:

- 1) 14
- 2) 15
- 3) 4
- 4) 9
- 5) 11

Задание #45

Вопрос:

$x := 5848;$

Выберите все выражения, которые имеют значение TRUE

Выберите несколько из 5 вариантов ответа:

- 1) $(x \bmod 100) < ((x \bmod 1000) \operatorname{div} 10)$
- 2) $((x \bmod 10) - ((x \operatorname{div} 100) \bmod 10)) = 0$
- 3) $((x \operatorname{div} 100) \bmod 10) / 2 = 4$
- 4) $(x \bmod 100) + (x \operatorname{div} 1000) = 50$
- 5) $((x \bmod 1000) \operatorname{div} 100) / (x \bmod 10) = 10$

Задание #46

Вопрос:

$$P = \begin{cases} x^2 & \text{при } x=1 \\ 2x^2-1 & \text{при } x=3 \\ |x-12| & \text{при } x=5 \\ x/2 & \text{в остальных случаях} \end{cases}$$

Выберите верный вариант использования CASE для решения задачи

Выберите один из 5 вариантов ответа:

- 1) Case x of
1: $p := \operatorname{sqr}(x);$
3: $p := 2 * \operatorname{sqr}(x) - 1;$
5: $p := \operatorname{abs}(x - 12);$
else $p := x / 2;$
end;
- 2) Case p of
1: $p := \operatorname{sqr}(x);$
3: $p := 2 * \operatorname{sqr}(x) - 1;$
5: $p := \operatorname{abs}(x - 12);$
else $p := x / 2;$
end;
- 3) Case x of
1: $p := \operatorname{sqr}(x);$
3: $p := 2 * \operatorname{sqr}(x) - 1;$
5: $p := \operatorname{abs}(x - 12);$
else $p := x / 2;$
- 4) Case x of
1: $p := \operatorname{sqr}(x);$
3: $p := 2 * \operatorname{sqr}(x) - 1;$
5: $p := \operatorname{abs}(x - 12);$
end;
 $p := x / 2;$

```
5) Case x of;
1: p:=2*sqr(x)-1;
3: p:=sqr(x);
5: p:=abs(x-12);
else p:=x/2;
end;
```

Задание #47

Вопрос:

Выберите верный вариант описания принципа работы оператора IF

Выберите один из 5 вариантов ответа:

- 1) Сначала проверяется условие, если условие истинно, то выполнится оператор, стоящий после THEN. Если условие ложно, то выполнится оператор, стоящий после ELSE.
- 2) Сначала проверяется условие, если условие истинно, то выполнится оператор, стоящий после ELSE. Если условие ложно, то выполнится оператор, стоящий после THEN.
- 3) Сначала выполняется оператор, стоящий после THEN. Затем, если условие верное, то выполнится оператор, стоящий после ELSE.
- 4) Сначала проверяется условие, если условие ложно, то выполнится оператор, стоящий после THEN. Если условие истинно, то выполнится оператор, стоящий после ELSE.
- 5) Сначала проверяется условие, если условие истинно, то выполнится оператор, стоящий после THEN. Если условие ложно, то выполнится оператор, стоящий после ELSE.

Задание #48

Вопрос:

Как правильно записать условие для

определения, какое число является большим из трех, введенных пользователем

Выберите один из 4 вариантов ответа:

- 1) (a>b) and (a>c)
- 2) (a>b) or (a>c)
- 3) c>a and c>b
- 4) (b>a and a>c)

Задание #49

Вопрос:

Сколько уровней вложений операторов IF можно использовать в программе?

Выберите один из 5 вариантов ответа:

- 1) 3
- 2) 5
- 3) 7
- 4) неограниченно
- 5) нельзя использование вложение операторов IF

Задание #50

Вопрос:

Переменные какого типа можно использовать при применении оператора CASE?

Выберите один из 7 вариантов ответа:

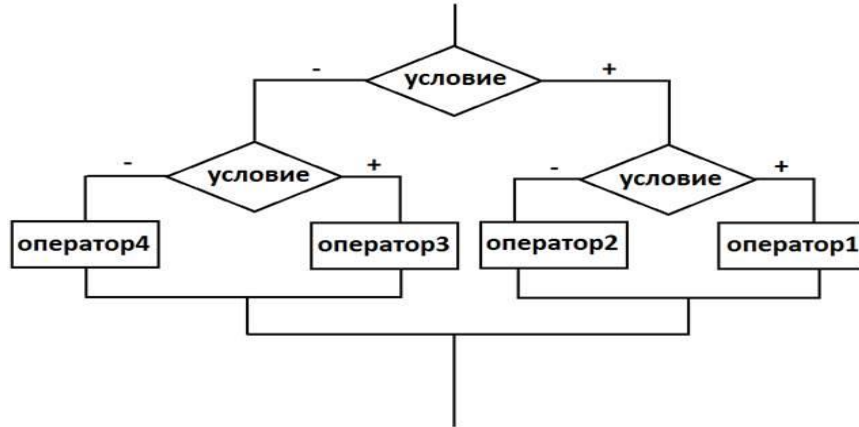
- 1) любой тип
- 2) word, char, boolean, byte
- 3) array, string, byte, word
- 4) char, integer, byte, string
- 5) byte, real, string, char
- 6) boolean, byte, array, real
- 7) real, char, boolean, byte

Задание #51

Вопрос:

Выберите правильный фрагмент программы, который соответствует блок-схеме (см. рис)

Изображение:



Выберите один из 4 вариантов ответа:

- 1) IF УСЛОВИЕ THEN IF УСЛОВИЕ THEN ОПЕРАТОР1 ELSE ОПЕРАТОР2 ELSE IF УСЛОВИЕ THEN ОПЕРАТОР3 ELSE ОПЕРАТОР4;
- 2) IF УСЛОВИЕ THEN IF УСЛОВИЕ THEN ОПЕРАТОР2 ELSE ОПЕРАТОР1 ELSE IF ОПЕРАТОР3 ELSE ОПЕРАТОР4;
- 3) IF УСЛОВИЕ THEN IF УСЛОВИЕ THEN ОПЕРАТОР1 ELSE ОПЕРАТОР2; ELSE IF УСЛОВИЕ THEN ОПЕРАТОР4 ELSE ОПЕРАТОР3;
- 4) IF УСЛОВИЕ IF УСЛОВИЕ THEN ОПЕРАТОР1 ELSE ОПЕРАТОР2 ELSE IF УСЛОВИЕ THEN ОПЕРАТОР3 ELSE ОПЕРАТОР4;

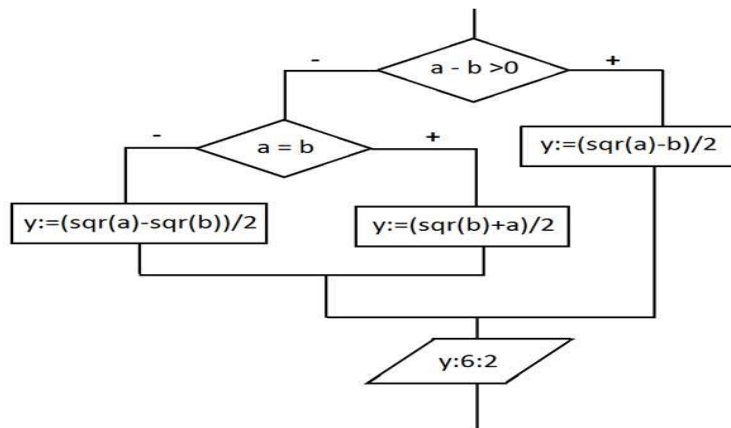
Задание #52

Вопрос:

Чему будет равен результат после выполнения следующего фрагмента (см. рис.)

Исходные данные: $a = 7, b = 7$

Изображение:



Выберите один из 5 вариантов ответа:

- 1) 28.00
- 2) 21.00
- 3) 7.00

- 4) 0
- 5) 28

Задание #53

Вопрос:

Установить соответствие

Укажите соответствие для всех 4 вариантов ответа:

- 1) FOR
- 2) REPEAT
- 3) WHILE
- 4) нет такого цикла

- цикл с параметром
- цикл с постусловием
- цикл с предусловием
- прямой цикл

Задание #54

Вопрос:

Какой цикл лучше использовать, если заранее известно, сколько раз будет выполняться циклическая часть?

Выберите один из 4 вариантов ответа:

- 1) цикл с параметром
- 2) цикл с постусловием
- 3) цикл с предусловием
- 4) фиксированный цикл

Задание #55

Вопрос:

Что означает следующая команда:

FloodFill(x,y,14)

Выберите один из 5 вариантов ответа:

- 1) Заливка графического объекта, контур которого желтого цвета
- 2) Заливка графического объекта желтым цветом
- 3) Рисование окружности, x,y - центр, 14 - радиус
- 4) Вывод точки красного цвета с координатами x,y
- 5) Заливка графического объекта, контур которого красного цвета

Задание #56

Вопрос:

Что такое алгоритм?

Выберите один из 4 вариантов ответа:

- 1) строгая последовательность действий, предназначенная для решения однотипных задач
- 2) последовательность действий, которая определяет этапы решения задачи на ПК
- 3) перевод программы, написанной на языке программирования на машинный язык
- 4) строгая последовательность действий, используемая для тестирования программы

Задание #57

Вопрос:

Что означает свойство алгоритма "массовость"?

Выберите один из 5 вариантов ответа:

- 1) с помощью алгоритма можно решать однотипные задачи

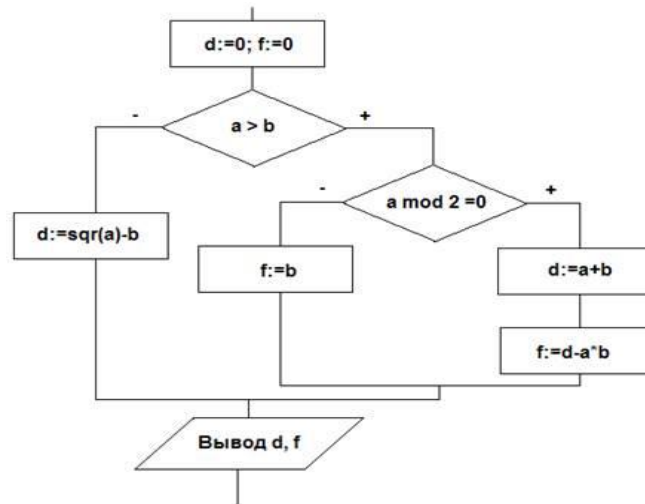
- 2) алгоритм должен быть разбит на простые шаги
- 3) после выполнения алгоритма обязательно должен быть получен результат
- 4) алгоритм должен быть составлен таким образом, чтобы его можно было легко перевести на язык программирования
- 5) алгоритм может быть частью большой программы

Задание #58

Вопрос:

Выберите правильный фрагмент программы, который соответствует блок-схеме. (см. рисунок)

Изображение:



Выберите один из 4 вариантов ответа:

- 1) d:=0; f:=0; If a>b then If a mod 2=0 then begin d:=a+b; f:=d-a*b; end else f:=b else d:=sqrt(a)-b; writeln(d,' ',f);
- 2) d:=0; f:=0; If a>b then If a mod 2=0 then begin d:=a+b; f:=d-a*b; end; else f:=b; else d:=sqrt(a)-b; writeln(d,' ',f);
- 3) d:=0; f:=0; If a<b then If a mod 2=0 then begin d:=a+b; f:=d-a*b; end else f:=b else d:=sqrt(a)-b; writeln(d,' ',f);
- 4) d:=0; f:=0; If a>b then If b mod 2=0 then begin d:=a+b; f:=d-a*b; end else f:=b else d:=sqrt(a)-b; writeln(d,' ',f);

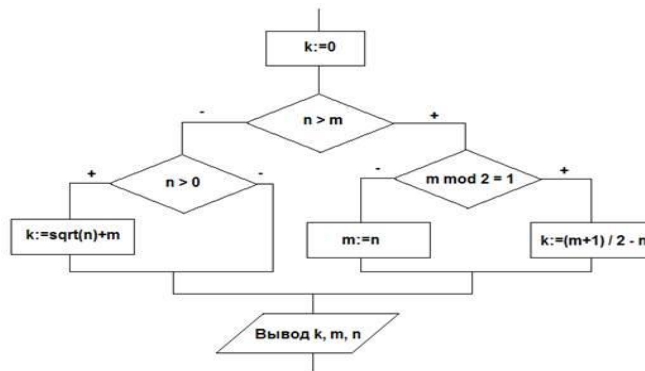
Задание #59

Вопрос:

Чему будет равен результат после выполнения следующего фрагмента программы при n=16, m=16? (см. рисунок)

Какие числа будут выведены в результате?

Изображение:



Выберите один из 5 вариантов ответа:

- 1) 20 16 16
- 2) 16 20 16
- 3) 0 16 16
- 4) 2 16 0
- 5) 0 0 0

Задание #60

Вопрос:

Выберите правильную форму записи оператора IF

Выберите один из 5 вариантов ответа:

- 1) IF условие THEN оператор1 ELSE оператор2;
- 2) IF оператор1 THEN условие ELSE оператор2;
- 3) IF условие ;THEN оператор1; ELSE оператор2;
- 4) IF условие ELSE оператор1 THEN оператор2;
- 5) IF условие THEN оператор2 ELSE оператор2;

Задание #61

Вопрос:

Определите по блок-схеме (см. рисунок), сколько раз выполнится циклическая часть

Изображение:

```
Program zadacha;  
Var q, w: integer;  
Begin  
q:=1; w:=0;  
Repeat  
  q:=q+1;  
  w:=w+q/2;  
Until q=8;  
WriteIn('w=',w:6:2);  
ReadIn;  
End.
```

Выберите один из 5 вариантов ответа:

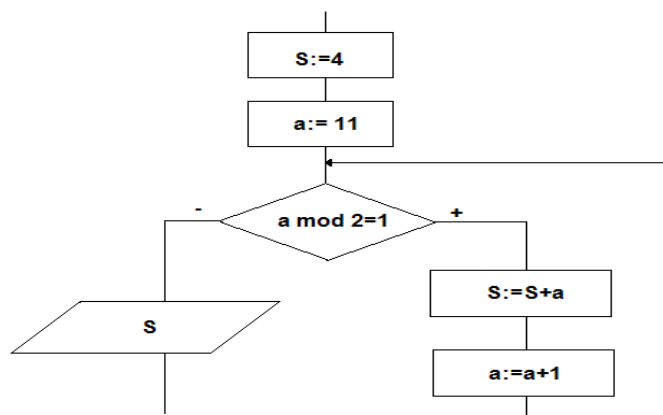
- 1) 7
- 2) 8
- 3) 6
- 4) 5
- 5) 0

Задание #62

Вопрос:

Определить по блок-схеме, чему будет результат S ? (см. рисунок)

Изображение:



Выберите один из 5 вариантов ответа:

- 1) 15
- 2) 11
- 3) 9.5
- 4) 4
- 5) -7

Задание #63

Вопрос:

Что такое составной оператор?

Выберите один из 5 вариантов ответа:

- 1) несколько операторов, которые объединены в группу
- 2) группа нескольких условий
- 3) совокупность арифметических и математических действий
- 4) выражение, в котором используются несколько переменных
- 5) несколько операторов, следующих друг за другом

Задание #64

Вопрос:

Задача. Дан одномерный массив из 15 элементов. Элементы - целые значения. Заполнение с помощью случайных чисел от -15 до +15.

Выберите правильный вариант описания массива.

Выберите один из 5 вариантов ответа:

- 1) Var W : array[1..15] of integer;
- 2) Const W : array[1..12] of integer;
- 3) Var W : array[-15..15] of integer;
- 4) Var W : array[1..15] of real;
- 5) Type W : array[1..15] of integer;

Задание #65

Вопрос:

Дан двумерный массив (изображен на рисунке).

Чему равен элемент В[3,2] ?

Изображение:

В

-10	2	5	6	-11	15
-9	0	-101	-2	3	100
22	4	16	-4	-13	13

Выберите один из 5 вариантов ответа:

- 1) -101
- 2) 0
- 3) 16
- 4) -2
- 5) 4

Задание #66

Вопрос:

Дан одномерный массив из 10 элементов. Как правильно определить максимальный элемент в массиве? Максимальный элемент только один.

Выберите правильный фрагмент программы.

Выберите один из 5 вариантов ответа:

- 1) max:=A[1]; For i:=1 to 10 do If A[i]>max then max:=A[i]; Writeln('max=',max);
- 2) max:=A[1]; For i:=1 to 10 do If A[i]>max then begin max:=A[i]; Writeln('max=',max);end;
- 3) max:=A[1]; For i:=1 to 10 do; If A[i]>max then max:=A[i]; Writeln('max=',max);
- 4) For i:=1 to 10 do If A[i]>max then max:=A[i]; max:=A[1]; Writeln('max=',max);
- 5) max:=A[1]; For A:=1 to 10 do If A[i]>max then max:=A[i]; Writeln('max=',max);

Задание #67

Вопрос:

Выберите правильный фрагмент программы, как вывести одномерный массив в виде столбца

Выберите один из 5 вариантов ответа:

- 1) For i:=1 to 16 do Write(D[i], ' ');
- 2) For i:=1 to 16 do Writeln(D[i]);
- 3) For i:=1 to 16 do; Write(D[i], ' '); Writeln;
- 4) For i:=16 downto 1 do Write(D[i], ' ');
- 5) For i:=1 to 16 do Write(D[i]);

Задание #68

Вопрос:

Выберите правильный фрагмент программы, как вычислить количество четных элементов в массиве. (ноль в качестве четного элемента не должен учитываться).

Выберите один из 5 вариантов ответа:

- 1) n:=0; For k:=1 to 20 do If (A[k] mod 2=0) and (A[k]<>0) then n:=n+1;
- 2) n:=0; For k:=1 to 20 do; If (A[k] mod 2=1) and (A[k]<>0) then n:=n+1;
- 3) n:=0; For k:=1 to 20 do If (A[k] mod 2=0) and (A[k]<>0) then n:=n-1;
- 4) n:=0; For k:=1 to 20 do If (A[n] mod 2=0) and (A[n]<>0) then n:=n+1;
- 5) n:=0; For k:=1 to 20 do If (A[k] mod 2=0) or (A[k]<>0) then n:=n+1;

Задание #69

Вопрос:

команда CIRCLE(120,15,150);

Что означает число "15" в записи команды?

Выберите один из 5 вариантов ответа:

- 1) обозначение цвета
- 2) радиус
- 3) значение координаты центра по оси y
- 4) значение координаты вершины дуги
- 5) значение координаты центра по оси x

Задание #70

Вопрос:

Выберите все верные высказывания:

Выберите несколько из 6 вариантов ответа:

- 1) для проверки нескольких условий используются логические операции (and, or, not)
- 2) если в записи условий используются and, or, not, то необходимо каждое условие поместить в отдельные скобки
- 3) когда используется оператор case, то в программе на один end больше
- 4) оператор case можно использовать для решения любых задач с условиями
- 5) в операторе IF использовать ELSE обязательно
- 6) количество операторов begin и end всегда должно быть одинаково

Задание #71

Вопрос:

Выберите все верные высказывания:

Выберите несколько из 7 вариантов ответа:

- 1) в цикл с параметром заранее известно, сколько раз будет выполняться циклическая часть
- 2) в цикл с предусловием заранее известно, сколько раз будет выполняться циклическая часть
- 3) в цикле с постусловием шаг может быть только целым значением
- 4) в цикле с параметром шаг может быть только целым значением
- 5) в цикле с предусловием не надо задавать начальное значение
- 6) цикл с параметром - for, цикл с предусловием - repeat, цикл с постусловием - while
- 7) в цикл с постусловием заранее неизвестно, сколько раз будет выполняться циклическая часть

Задание #72

Вопрос:

Выберите все правильные идентификаторы.

Выберите несколько из 8 вариантов ответа:

- 1) текст
- 2) text
- 3) a#2
- 4) a_2
- 5) a2
- 6) randoize
- 7) a\$
- 8) kolichestvo

Задание #73

Вопрос:

Выберите все неправильные идентификаторы.

Выберите несколько из 8 вариантов ответа:

- 1) writeln
- 2) writeln_1
- 3) a#2
- 4) 1a
- 5) abs
- 6) abc
- 7) summa
- 8) kolichestvo

Задание #74

Вопрос:

Расставьте в правильном порядке строки программы для решения задачи:

Укажите порядок следования всех 10 вариантов ответа:

- ___ Program odin;
- ___ Uses crt;

```

__ Var a,b:real;
__ begin
__ writeln('a-?');
__ readln(a);
__ b:=a/2;
__ readln;
__ end.
__ writeln('b=',b:6:2);

```

Задание #75

Вопрос:

Установите соответствие между значением и типом данных

Укажите соответствие для всех 6 вариантов ответа:

- 1) integer
- 2) real
- 3) word
- 4) longint
- 5) byte
- 6) число так записывать нельзя

```

__ -851
__ 12
__ 1254
__ 69541
__ 23.457
__ 23,547

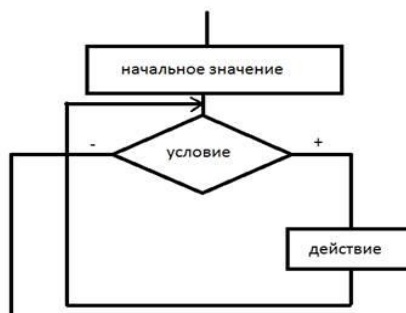
```

Задание #76

Вопрос:

Какой вид цикла изображен на рисунке?

Изображение:



Выберите один из 5 вариантов ответа:

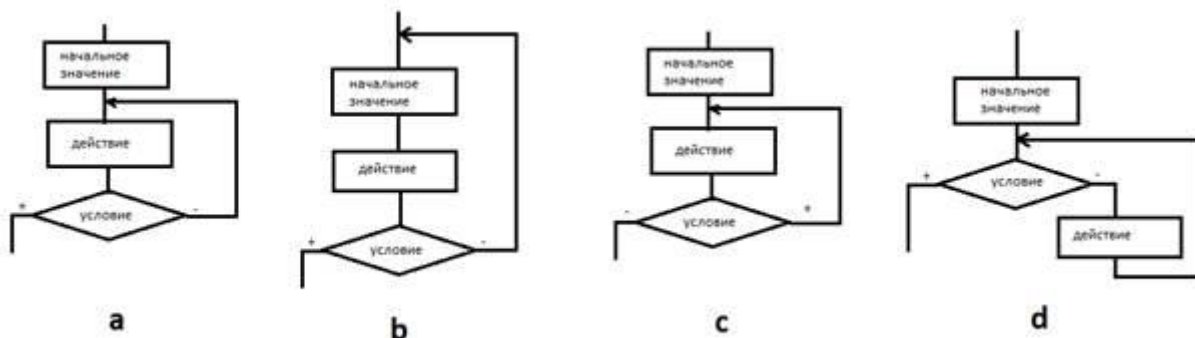
- 1) цикл с параметром
- 2) цикл с предусловием
- 3) цикл с постусловием
- 4) блок-схема составлена неверно
- 5) простой цикл

Задание #77

Вопрос:

Выберите верный вариант блок-схемы цикла с постусловием (см. рис.)

Изображение:



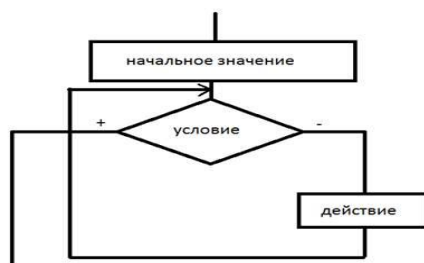
Запишите ответ:

Задание #78

Вопрос:

Какой вид цикла изображен на рисунке?

Изображение:



Выберите один из 5 вариантов ответа:

- 1) цикл с параметром
- 2) цикл с предусловием
- 3) цикл с постусловием
- 4) блок-схема составлена неверно
- 5) простой цикл

Задание #79

Вопрос:

Чем отличается процедура от функции?

Выберите один из 5 вариантов ответа:

- 1) ничем
- 2) функция возвращает значение в программу
- 3) процедура возвращает значение в программу
- 4) нельзя использовать функцию без параметров
- 5) количеством обрабатываемых параметров

Задание #80

Вопрос:

Выберите из приведенных примеров ФУНКЦИИ

Выберите несколько из 7 вариантов ответа:

- 1) clrscr
- 2) sqrt
- 3) readln

- 4) random
- 5) randomize
- 6) program
- 7) copy

Задание #81

Вопрос:

Выберите из приведенных примеров ПРОЦЕДУРЫ

Выберите несколько из 7 вариантов ответа:

- 1) readln
- 2) randomize
- 3) abs
- 4) random
- 5) str
- 6) sqr
- 7) copy

Задание #82

Вопрос:

Дано слово: a:='календарь';

Какое слово получится в результате выполнения следующей операции:

b:=copy(a,3,2)+copy(a,1,2)+copy(a,8,2);

Запишите ответ:

Задание #83

Вопрос:

Дано слово: a:='электричество';

Какое слово получится в результате выполнения следующей операции:

b:=copy(a,4,1)+copy(a,6,1)+copy(a,9,2)+copy(a,2,1)+copy(a,13,1);

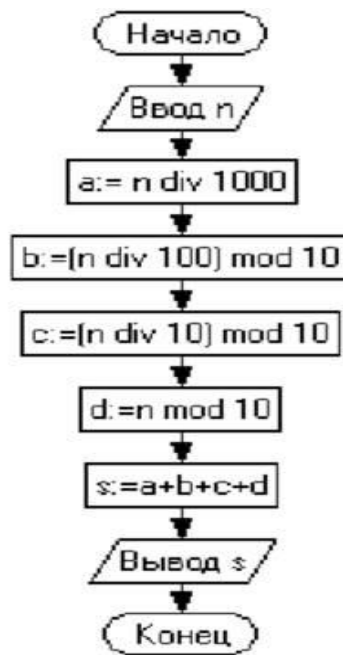
Запишите ответ:

Задание #84

Вопрос:

Алгоритм задан блок-схемой.

Чему будет равен результат, если n= 5874 ?



Запишите число:

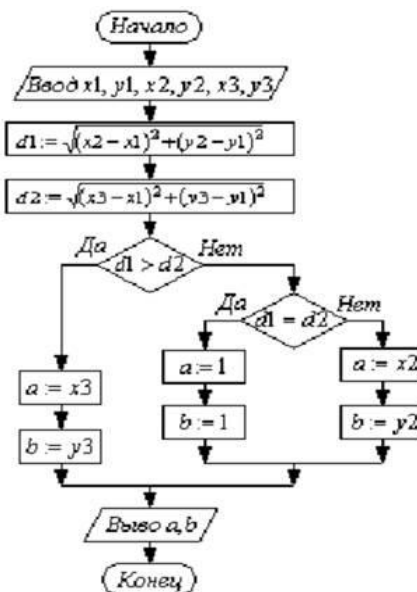
Задание #85

Вопрос:

Алгоритм задан блок-схемой.

В результате выполнения алгоритма при исходных данных $x_1=2, y_1=-1, x_2=-1, y_2=-2, x_3=2, y_3=2$

Чему равно значение переменной b?

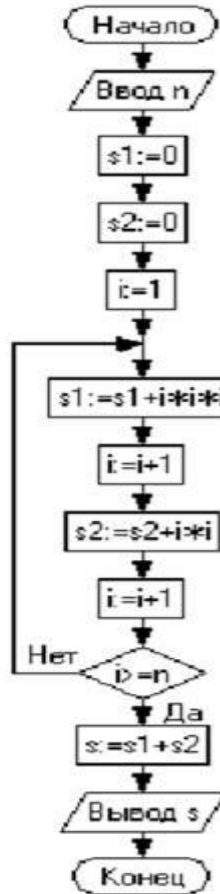


Запишите число:

Задание #86

Вопрос:

Чему будет равно s , если $n=6$?

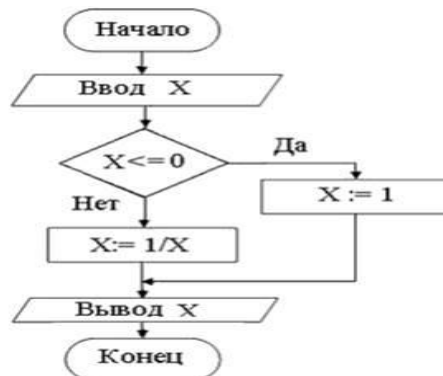


Запишите число:

Задание #87

Вопрос:

Какое значение выведется в результате выполнения данного алгоритма?



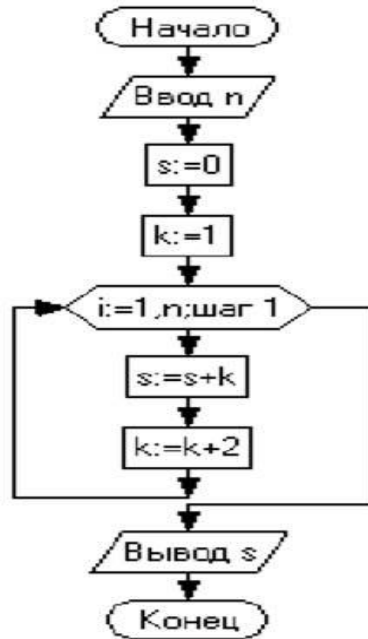
Выберите один из 4 вариантов ответа:

- 1) 1, если вводится отрицательное значение и значение $1/X$ в остальных случаях
- 2) $1/X$ для любого вводимого значения X
- 3) 1, если вводится и значение $1/X$ в остальных случаях
- 4) X , если вводится положительное значение и значение $1/X$ в остальных случаях

Задание #88

Вопрос:

Чему будет равно s , если $n=7$?



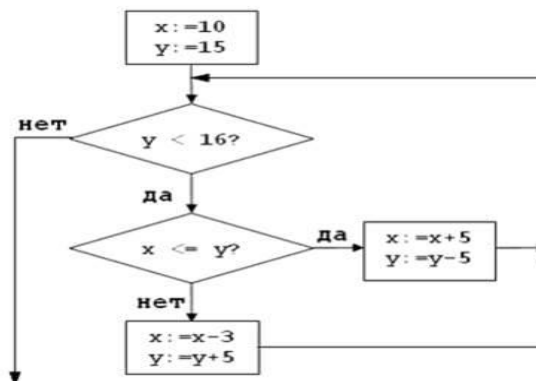
Выберите один из 4 вариантов ответа:

- 1) 49
- 2) 28
- 3) 36
- 4) 54

Задание #89

Вопрос:

В ниже приведенной схеме для организации цикла с предусловием используется(-ются) ...



Выберите один из 4 вариантов ответа:

- 1) блок «Решение» с условием $y < 16$
- 2) блок «Решение» с условием $x \leq y$
- 3) оба блока «Решение» с условиями $x \leq y$ и $y < 16$
- 4) блок «Процесс» $x := 10$

Задание #90

Вопрос:

Дана последовательность действий:

- 1) просыпаемся;
- 2) смотрим на часы;
- 3) бросаем монетку;
- 4) если выпадает «орел», то идем на занятия, если выпадает «решка» - спим дальше.

Тогда такая последовательность действий может быть описана алгоритмической структурой, которая называется ...

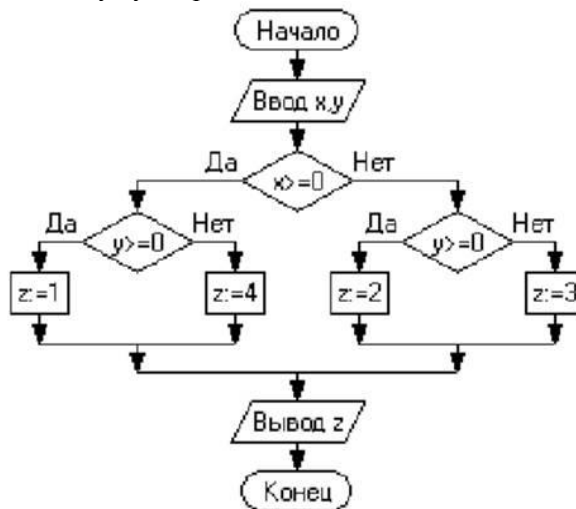
Выберите один из 4 вариантов ответа:

- 1) ветвление
- 2) цикл с параметром
- 3) цикл с предусловием
- 4) линейная

Задание #91

Вопрос:

Исходные данные: $x=-25$, $y=1$. Чему будет равно значение z ?



Запишите число:

Задание #92

Вопрос:

Компиляция и интерпретация представляют собой ...

Выберите один из 4 вариантов ответа:

- 1) два принципиально различных метода трансляции
- 2) два синонима для обозначения процесса перевода программного кода с языка программирования высокого уровня на машинный язык и создания исполняемого файла
- 3) два синонима для обозначения процесса построения и выполнения программного кода
- 4) два этапа загрузки операционной системы

Задание #93

Вопрос:

В языке программирования Pascal оператор GOTO является оператором ...

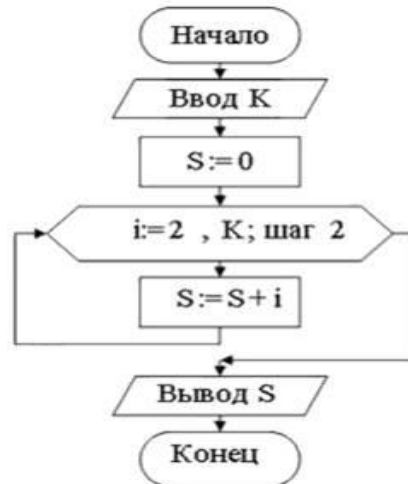
Выберите один из 5 вариантов ответа:

- 1) безусловного перехода
- 2) условного перехода
- 3) ветвления
- 4) выбора
- 5) цикла

Задание #94

Вопрос:

Что будет вычислено в результате данного алгоритма при вводе натурального K?



Выберите один из 4 вариантов ответа:

- 1) суммы четных чисел среди первых натуральных K чисел
- 2) суммы нечетных чисел среди первых натуральных K чисел
- 3) суммы первых натуральных K чисел
- 4) суммы первых натуральных K-1 чисел

Задание #95

Вопрос:

Что будет выведено в результате, если B=26?



Выберите один из 4 вариантов ответа:

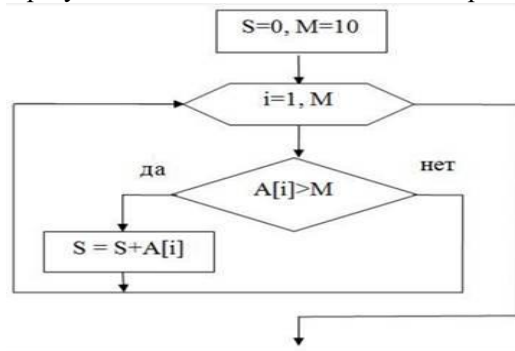
- 1) работник
- 2) пенсионер
- 3) учащийся
- 4) дошкольник

Задание #96

Вопрос:

Тема "Массивы"

Чему будет равно значение S в результате выполнения данного алгоритма?



Выберите один из 4 вариантов ответа:

- 1) сумме десяти элементов массива
- 2) наибольшему элементу массива
- 3) сумме элементов, значения которых больше 10
- 4) сумме элементов, индексы которых больше 10

Задание #97

Вопрос:

Тема "Массивы"

Составляющая массива называется ... массива.

Выберите один из 5 вариантов ответа:

- 1) элемент
- 2) индекс
- 3) описание
- 4) переменная
- 5) параметр

7.4. Оценочные средства промежуточной аттестации

7.4.1. Примерные вопросы

Вопрос 1

Главный объект проекта Delphi называется

Варианты ответов

- Application
- Form
- Properties
- Object Inspector

Вопрос 2

Каково назначение компонента label

Варианты ответов

- Ввод исходных данных
- Запуск приложения
- Вывод текстовой информации.
- Вывод надписи на командной кнопке

Вопрос 3

Свойством формы являются

Варианты ответов

- Color
- Items
- Caption
- Lines

Вопрос 4

В Инспекторе объектов находятся две закладки

Варианты ответов

- свойства и надписи
- свойства и методы
- список файлов и компонентов
- команды меню

Вопрос 5

Для сохранения всего проекта в Delphi используется команда

Варианты ответов

- Save
- Save As...
- Save All
- Save Project

Вопрос 6

Для ввода исходных данных в Delphi служит компонент

Варианты ответов

- Edit
- Button
- Label
- Memo

Вопрос 7

Найдите соответствие

Варианты ответов

- Компонент Edit
- Компонент Button
- Компонент Memo

Вопрос 8

Найдите соответствие

Варианты ответов

- Компонент ComboBox
- Компонент CheckBox
- Компонент RadioButton

7.4.2. Примерные задачи

Контрольные тесты

по теме «Объектно-ориентированное программирование в среде Delphi»

Вариант 1

1. **Методика разработки программ, в основе которой лежит понятие объект.**

1. Структурное программирование.
2. Процедурное программирование
3. Объектно-ориентированное программирование
4. Модульное программирование

2. **Функция преобразования строки в вещественное число**

1. StrToFloat (Строка):Extended
2. Case RadioGroup1.ItemIndex of
3. b:=StrToInt(Edit2.text);
4. Edit3.text:=

1. **Обработчик события - это фрагмент**

0. представляет собой последовательность текстовых строк
1. представляет собой последовательность текстовых строк, в которых указывается, что именно должна делать программа в ответ на нажатие кнопки
2. программный код
3. программный модуль

1. **Вывод данных производится:**

- A) В окно сообщения, в текстовый файл, открытый для записи
B) В окно сообщения, в поле диалогового окна, в текстовый файл, открытый для записи
C) В поле диалогового окна, в текстовый файл, открытый для записи
D) В окно сообщения, в поле диалогового окна,

1. **В Delphi укажите инструкцию цикла?**

0. REPEAT... UNTIL
1. CASE OF
2. IF <условие> THEN <инс.1> ELSE <инс.2>
3. GOTO N

2. **Категория объектов, обладающих одинаковыми свойствами и поведением это...**

0.
 0. Класс.
 1. Тип.
 2. Структура.
 3. Система.

3. **Событие – это..**

0. Изменение состояния объекта в ответ на какое-либо
1. Совокупность объектов, характеризующаяся общностью методов и свойств
2. Действие, которое может выполнить объект
3. Характеристика объекта

1. **Свойство – это...**

1.
 0.
 0. Характеристика объекта
 1. Совокупность объектов, характеризующаяся общностью методов и свойств
 2. Изменение состояния объекта в ответ на какое-либо
 3. Действие, которое может выполнить объект

1. **На какой вкладке Объектного инспектора Object Inspector отображаются свойства компонентов**

0. Unit

1. Properties
2. Events
3. TreeView
2. **Свойство формы Top – это...**
 0. расстояние от верхней границы формы до верхней границы экрана
 1. имя формы, используется для управления формой и доступа к компонентам формы
 2. текст заголовка
 3. ширина формы



3. **Предназначен данный элемент управления: TMainMenu ?**
 0. Позволяет создавать всплывающие меню
 1. Позволяет выбирать только одну опцию из нескольких.
 2. Используется для представления иерархических отношений связанных данных
 3. Позволяет поместить главное меню в программу




4. **Что представляет собой данный элемент управления: TEdit ?**
 0. Стандартный управляющий элемент для ввода данных, который может быть использован для отображения короткого фрагмента текста и позволяет пользователю вводить текст во время выполнения программы.
 1. Специализированный ComboBox для выбора текущего диска.
 2. Специализированный список, в котором отображаются файлы из указанной директории
 3. Элемент оформления для создания заголовков с изменяемыми размерами для таблиц.



5. **Для чего предназначен данный элемент управления: TCheckBox ?**
 0. Отображает строку текста с маленьким окошком рядом, в котором можно поставить отметку, которая означает, что что-то выбрано.
 1. Полоса прокрутки, появляющаяся автоматически в объектах редактирования
 2. Элемент оформления для создания заголовков с изменяемыми размерами для таблиц.
 3. Отображает графическое изображение на форме.
6. **Свойство ItemIndex компонента TListBox - ...**
 0. Выравнивает компонент на форме
 1. Указывает порядковый номер элемента, выделенного в списке
 2. Содержит текст, отображаемый в окне подсказки или в окне состояния
 3. Массив строк списка - объект класса TStringList. Свойства этого класса позволяют формировать и изменять список
7. **На какой странице палитры компонентов находится компонент TImage?**
 0. Additional
 1. Standard
 2. Win32
 3. System
8. **Свойство Picture компонента TImage- ...**

- 0. Указывает, должны ли изменяться размеры изображения, подгоняясь под размеры компонента
 - 1. Указывает, должен ли быть цвет фона изображения прозрачным, чтобы сквозь него было видно нижележащее изображение
 - 2. Указывает, изменяется ли автоматически размер компонента, подстраиваясь под размер изображения
 - 3. Определяет отображаемый графический объект типа TPicture. Может загружаться программно или во время проектирования
9. **Укажите какой из перечисленных компонентов содержит страница Standard палитры компонентов в Delphi?**
- 0. TImage
 - 1. TMaskEdit
 - 2. TPanel
 - 3. TShape
10. **Массив – это...**
- 0. это набор данных, следующих друг за другом.
 - 1. Возникшая в процессе выполнения программы ошибка
 - 2. Реакция приложения на операцию ввода
 - 3. Используемая в процессе написания программы инструкция, для обозначения комментария
11. **Что произойдет после выполнения этой процедуры?**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
form1.Color:=clred;
end;
```
- 0. Откроется окно приложения.
 - 1. Закроется окно приложения.
 - 2. Изменится цвет окна приложения.
 - 3. Изменится заголовок окна приложения.
12. **Какие объекты из перечисленных не имеют свойства Caption?**
- 0. RadioButton1.
 - 1. Image1.
 - 2. CheckBox1.
 - 3. Edit1.
13.  **Какое событие может быть связано с данным элементом управления?**
- 0. OnClick.
 - 1. OnChange.
 - 2. OnTimer.
 - 3. OnExit.
14. **Что произойдет после выполнения этой процедуры?**

```
procedure TForm1.Button1Click(Sender: TObject);
begin
form1.Close;
end;
```
- 1. Откроется окно приложения.
 - 2. Закроется окно приложения.
 - 3. Изменится цвет окна приложения.
 - 4. Изменится заголовок окна приложения.

1. **Какое свойство отвечает за видимость/невидимость компонента на форме?**
 1.
 0. Align
 1. Caption
 2. Visible
 3. Name
24. **Какая страница палитры компонентов в в Delphi содержит компоненты TSpeedButton, TImage, TShape, TMaskEdit, TStringGrid и т.д.**
 1. Additional
 2. Standard
 3. Win32
 4. System
1. **Поверхность, на которую можно вывести графику - ...**
 1.
 0. Color
 1. Canvas
 2. Font
 3. Icon

Вариант 2

1. **Объектно-ориентированное программирование – это**
 1.
 0. применение знания языка программ.Pascal для программирования в среде Delphi
 1. методика разработки программ, в основе которой лежит понятие объект.
 2. формирование научного мировоззрения, привитие интереса к предмету.
 3. процесс конструирования визуального проявления программы
2. **Функция преобразования текстовой строки в целое число**
 1. StrToInt (Строка):integer
 2. Case RadioGroup1.ItemIndex of
 3. b:=StrToInt(Edit2.text);
 4. Edit3.text:=";
3. **Процесс создания Delphi-программы разбивается на фазы:**
 1. фазы проектирования, конструирования и кодирования;
 2. фазу проектирования формы и фазу кодирования;
 3. фазу конструирования формы и фазу кодирования;
 4. фазу конструирования и кодирования формы;
4. **Ввод данных осуществляется:**
 1. из поля окна диалога, из текстового файла
 2. из диалогового окна сообщения, из поля окна диалога
 3. из диалогового окна сообщения, из текстового файла
 4. из диалогового окна сообщения, из поля окна диалога, из текстового файла
5. **В Delphi укажите инструкцию условия?**
 1. REPEAT... UNTIL
 2. WHILE <логическое выражение>_ DO
 3. IF <условие> THEN <инс.1> ELSE <инс.2>
 4. GOTO N
6. **Программный объект включает в себя...**
 1. Несколько модулей.

2. Несколько процедур по обработке данных.
3. Данные и процедуры по обработке этих данных.
4. Форму и расположенные на ней элементы управления.

1.

7. Класс – это...

1. Совокупность объектов, характеризующаяся общностью методов и свойств
2. Изменение состояния объекта в ответ на какое-либо
3. Действие, которое может выполнить объект
4. Характеристика объекта

8. Метод – это...

1.

0. Действие, которое может выполнить объект

1. Совокупность объектов, характеризующаяся общностью методов и свойств
2. Изменение состояния объекта в ответ на какое-либо
3. Характеристика объекта

1. На какой вкладке Объектного инспектора Object Inspector отображаются события компонентов

0. TreeView
1. Events
2. Properties
3. Unit

10. Что является основой программы?

0. метка
1. компонент
2. форма
3. формат



11. Для чего предназначен данный элемент управления: TPopupMenu ?

0. Позволяет выбирать только одну опцию из нескольких.
1. Используется для показа прокручиваемого списка и позволяет вводить информацию в маленьком поле ввода сверху
2. Позволяет создавать всплывающие меню
3. Используется для представления иерархических отношений связанных данных.



12. Для чего предназначен данный элемент управления: TLabel ?

0. Для отображения текста на экране.
1. Для быстрой отмены выбора какого-либо объекта
2. Для представления данных любого типа в виде таблицы
3. Для отображения простейших графических объектов на форме: окружность, квадрат и т.п.



13. Для чего предназначен данный элемент управления: TRadioButton ?

0. Позволяет выбирать только одну опцию из нескольких.
1. Рисунок на форме

2. Используется для показа прокручиваемого списка и позволяет вводить информацию в маленьком поле ввода сверху
 3. Используется для представления иерархических отношений связанных данных.
14. **Свойство `ItemIndex` компонента `TComboBox` - ...**
0. Выравнивает компонент на форме
 1. Указывает номер выбранного элемента
 2. Содержит текст, отображаемый в окне подсказки или в окне состояния
 3. Массив строк списка - объект класса `TStrings`. Свойства этого класса позволяют формировать и изменять список
15. **На какой странице палитры компонентов находится компонент `TBitBtn`?**
1. Additional
 2. Standard
 3. Win32
 4. System
16. **Свойство `Stretch` компонента `TImage`- ...**
1. Указывает, должны ли изменяться размеры изображения, подгоняясь под размеры компонента
 2. Определяет отображаемый графический объект типа `TPicture`. Может загружаться программно или во время проектирования
 3. Указывает, должен ли быть цвет фона изображения прозрачным, чтобы сквозь него было видно нижележащее изображение
 4. Указывает, изменяется ли автоматически размер компонента, подстраиваясь под размер изображения
17. **Укажите какой из перечисленных компонентов содержит страница `Standard` палитры компонентов в `Delphi`?**
1. `TButton`
 2. `TImage`
 3. `TMaskEdit`
 4. `TShape`
18. **Набор данных, следующих друг за другом.**
1.
 0. массив
 1. исключение
 2. комментарий
 3. событие
19. **Что произойдет после выполнения этой процедуры?**
- ```

procedure TForm1.Button1Click(Sender: TObject);
begin
 form1.Caption:="red";
end;

```
1. Откроется окно приложения.
  2. Закроется окно приложения.
  3. Изменится цвет окна приложения.
  4. Изменится заголовок окна приложения.
20. **`Button1.Enabled:=false;` Данная инструкция делает элемент командную кнопку**
1.
    0. Видимой.
    1. Невидимой.
    2. Доступной.
    3. Недоступной.
21. **Какие из приведенных компонентов не являются оконными?**

- A)  . B)  . C)  . D)  .

22. Что произойдет после выполнения этой процедуры?

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
form1.Show;
```

```
end;
```

1.
  0. Откроется окно приложения.
  1. Закроется окно приложения.
  2. Изменится цвет окна приложения.
  3. Изменится заголовок окна приложения.
2. **Свойство Visible отвечает за...**
  0. прозрачность/непрозрачность компонента на форме
  1. активность/неактивность компонента на форме
  2. выравнивание компонента на форме
  3. видимость/невидимость компонента на форме
3. **Какая страница палитры компонентов в Delphi содержит стандартные компоненты Label, Button, Frames, Edit, Memo и т.д.**
  0. Standard
  1. Additional
  2. Win32
  3. System
4. **Свойство формы Canvas - ...**
  0. текст заголовка
  1. поверхность, на которую можно вывести графику
  2. ширина формы
  3. значок в заголовке диалогового окна, обозначающий кнопку вывода системного меню

### Примерные билеты к экзамену по дисциплине «Основы алгоритмизации и программирования»

#### Билет № 1

1. Что такое переменная?

2. Что из последовательностей символов является выражением, записанным по правилам Паскаля?

1)  $\sin(x) + \text{abs}(\text{abs}(y - 1.7))$ ;

2)  $x.8 + y^2$ ;

3)  $|h|/2$ ;

4)  $8 * a \& b$ ; 5)  $-5a$

#### Билет № 2

1. Что такое идентификатор?

2. Указать ошибки во фрагменте программы

```
program m5;
```

```
var f,g:real;
```

```
d:integer;
```

```
begin
```

```
d:=f mod g;
```

d:=f div g

end.

- 1) неверные имена программы и переменных;
- 2) неверный оператор присваивания;
- 3) переменная f должна быть типа integer, чтобы не возникло ошибки несовместимости типов;
- 4) функции mod и div работают только с переменными целого типа, результат тоже является целым;
- 5) переменная d должна быть типа real, чтобы не возникло ошибки несовместимости типов

#### Билет № 3

1. Какая команда служит для ввода данных? Для вывода данных?
2. Указать ошибки во фрагменте программы program m1; var a,b:real; begin b:=read(a) end;
  - 1) неверный оператор присваивания;
  - 2) переменная b должна быть типа integer, чтобы не возникло ошибки несовместимости типов;
  - 3) инструкцию read нельзя использовать в операторе присваивания;
  - 4) инструкция read работает только с переменными целого типа;
  - 5) переменная a должна быть типа integer, чтобы не возникло ошибки несовместимости типов.

#### Билет № 4

1. Чем отличается вывод информации на экран компьютера оператором write( ) от вывода оператором writeln( ) ?
2. Выражение  $|\sin x + \operatorname{tg} x|$  в Паскале должно быть записано в виде:
  - 1)  $|\sin(x)+\tan(x)|$  ;            2)  $\operatorname{abs}(\sin x+\operatorname{tg} x)$
  - 2)    3)  $\operatorname{abs}(\sin(x)+\tan(x))$  ;    4)  $\operatorname{abs}(\sin(x)+\sin(x)/\cos(x))$

Дано натуральное число, меньшее 9999. Получить новое число путём перестановки цифр в обратном порядке

#### Билет № 5

1. Как вывести на экран текст?
  2. Функция вычисления квадратного корня в Паскале
    - 1) SQR ;    2) RADICAL;    3) SQRT;    4) SQT
- Дано натуральное число, меньшее 9999. Получить новое число путём перестановки цифр в обратном порядке

#### Билет № 6

1. Можно ли вывести на экран несколько переменных или текстов одним оператором writeln()?
  2. Значение выражения  $3 \bmod 3$  равно...
- Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления среднего значения нечетных положительных чисел

### Билет № 7

1. Какие символы используются для размещения комментариев в программе?
  2. Значение выражения  $5 \text{ div } 2$  равно...
- Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления произведения чисел кратных 7

### Билет № 8

1. Чем отличается операция  $/$  от операции  $\text{div}$ ?
  2. Функция возведения в квадрат в Паскале
- 1) SQR; 2) INT ; 3) SQRT ; 4) SQT
- Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления суммы отрицательных чисел

### Билет № 9

1. Есть ли в *Pascal* операция возведения в степень?
  2. Выражение  $2(x-1) + \text{tg } x$  в Паскале должно быть записано в виде:
- 1)  $2(x-1)+\text{tan}(x)$ ; 2)  $2(x-1)+\text{tg}(x)$ ; 3)  $2*(x-1)+\text{tan}(x)$ ; 4)  $2*(x-1)+\text{sin}(x)/\text{cos}(x)$
- Даны 4 вещественных числа: a, b, c, d. Пройдёт ли кирпич с рёбрами a и b в прямоугольное отверстие в стене со сторонами c и d.

### Билет № 10

1. Придумайте формулу для нахождения предпоследней цифры числа?
  2. Выражение  $|5(\sin x + 2) - x|$  в Паскале должно быть записано в виде:
- 1)  $|5(\sin(x) + 2) - x|$ ; 2)  $|5*(\sin x + 2) - x|$   
3)  $\text{abs}(5(\sin(x) + 2) - x)$ ; 4)  $\text{abs}(5*(\sin(x) + 2) - x)$
- Даны 4 вещественных числа: a, b, c, d. Пройдёт ли кирпич с рёбрами a и b в прямоугольное отверстие в стене со сторонами c и d.

### Билет № 11

1. Можно ли производить операции mod и div над переменными, которые описаны как real?
  2. Выражение  $|5(\sin x + 2) - x|$  в Паскале должно быть записано в виде:
- 2)  $|5(\sin(x) + 2) - x|$ ; 2)  $|5*(\sin x + 2) - x|$   
4)  $\text{abs}(5(\sin(x) + 2) - x)$ ; 4)  $\text{abs}(5*(\sin(x) + 2) - x)$

Подсчитать количество пробелов, если в строке символов записано четверостишие А. Ахматовой «Калитку в милый сад, где клён, и дуб, и ясень, гуляя наугад, уж распахнула осень»

### Билет № 12

1. Что такое составное условие?
  2. Переменные целого типа описываются служебным словом...
- 1) Boolean 2) integer 3) byte 4) real 5) word 6) char  
7) longint 8) file 9) shortint 10) string

### Билет № 13

1. В чём отличие составного условия от простого?
2. Чем отличается вывод информации на экран компьютера оператором `write()` от вывода оператором `writeln()`

Составьте программу на Паскале, которая вставит в четверостишие А. Игнатовой «Сегодня такое спокойное море! Спокойно качаются в лодке буйки. ... лодка идёт на моторе, а в лодке спокойно сидят рыбаки» вместо «...» слово «Рыбацкая»

### Билет № 14

1. Объясните исполнение оператора `if`.
2. Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления. Суммы положительных чисел

### Билет № 15

1. Какие союзы используются при написании составного условия?
2. Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления суммы отрицательных чисел

### Билет № 16

1. В каких случаях после служебных слов `then` и `else` нужно писать `begin ... end`, а в каких нет?
2. Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления произведения чисел, кратных 7.

### Билет № 17

1. Что представляет собой составной оператор?
2. Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления количества чётных чисел

### Билет № 18

1. Почему `end` желательно записывать под `begin`?
2. Дан ряд чисел  $\{-25;25\}$ . Составьте программу для вычисления среднего значения нечетных положительных чисел

### Билет № 19

1. Для чего предназначен оператор цикла?
2. Даны 2 вещественных положительных числа  $r_1$  и  $r_2$ . Найти площадь кольца, образованного окружностями с радиусами  $r_1$  и  $r_2$

### Билет № 20

1. Какой формат записи имеет оператор `FOR`?
2. Как организовать цикл, чтобы получить ряд значений с шагом, отличным от 1. Например, с шагом 2 или -3?

### Билет № 21

1. Что такое элемент массива?

панели и панели сообщений, контейнеры

2. Даны 4 вещественных числа  $a, b, c, d$ . Определить, может ли прямоугольник со сторонами  $c$  и  $d$  поместиться в прямоугольнике  $a$  и  $b$ .

#### Билет № 22

1. Что такое индекс массива?

2. Как организовать цикл, чтобы получить ряд значений с шагом, отличным от 1.

Например, с шагом 2 или -3?

#### Билет № 23

1. Что такое размерность массива?

2. Перечислите части, из которых состоит программа на Паскале.

Составьте программу, которая создаст файл из 10 целых чисел, а потом выведет на экран 7-е, 6-е, 5-е, 4-е и 3-е числа.

#### Билет № 24

1. Зачем нужны массивы?

2. Дано натуральное число  $N < 999$ . Найти сумму цифр этого числа.

#### Билет № 25

1. Как можно обратиться к ячейке массива?

2. Дана строка символов  $s$ , символ  $c$ . Найти количество вхождений символа  $c$  в строку  $s$ .

Пример: Строка  $s = \text{'мама мыла раму'}$ , символ  $c = \text{'a'}$

#### Билет № 26

1. Какого типа могут быть элементы массива?

2. Даны 2 одномерных массива  $x$  и  $y$ . Найти числа, которые есть в обоих массивах

#### Билет № 27

1. Какого типа может быть индекс массива?

2. Даны 4 вещественных числа:  $a, b, c, d$ . Пройдёт ли кирпич с рёбрами  $a$  и  $b$  в прямоугольное отверстие в стене со сторонами  $c$  и  $d$ .

#### Билет № 28

1. Как можно осуществить ввод элементов массива?

2. Даны значения 3-х вещественных переменных  $a, b, c$ . Распределить их в порядке возрастания



## 8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Лекция 6. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Массивы. Двумерные массивы

### ОП.04 «Основы алгоритмизации и программирования»

Лекция 6. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Массивы. Двумерные массивы

#### Цели:

- 1) Актуализация полученных в курсе информатики знаний по ЯП Паскаль
- 2) Отработка навыков применения операторов цикла

#### Задачи:

- 1) повторить форматы операторов;
- 2) повторить понятие одномерного и двухмерного массивов;
- 3) актуализировать навыки применения операторов цикла к обработке массивов

#### Новый материал.

При решении задач может возникнуть необходимость повторить одни и те же действия несколько или множество раз. В программировании блоки кода, которые требуется повторять не единожды, оборачиваются в специальные конструкции – *циклы*. У циклов выделяют заголовок и тело. Заголовок определяет, до каких пор или сколько раз тело цикла будет выполняться. Тело содержит выражения, которые выполняются, если в заголовке цикла выражение вернуло логическую истину (True, не ноль). После того как достигнута последняя инструкция тела, поток выполнения снова возвращается к заголовку цикла. Снова проверяется условие выполнения цикла. В зависимости от результата тело цикла либо повторяется, либо поток выполнения переходит к следующему выражению после всего цикла.

В языке программирования Паскаль существует три вида циклических конструкций.



#### Цикл for

Часто цикл for называют циклом со счётчиком. Этот цикл используется, когда число повторений не связано с тем, что происходит в теле цикла. Т.е. количество повторений может быть вычислено заранее (хотя оно не вычисляется).

В заголовке цикла указываются два значения. Первое значение присваивается так называемой переменной-счетчику, от этого значения начинается отсчет количества итераций (повторений). Отсчет идет всегда с шагом равным единице. Второе значение

указывает, при каком значении счетчика цикл должен остановиться. Другими словами, количество итераций цикла определяется разностью между вторым и первым значением плюс единица. В Pascal тело цикла не должно содержать выражений, изменяющих счетчик.

Цикл for существует в двух формах:

```
for счетчик:=значение to конечное_значение do
```

```
 тело_цикла;
```

```
for счетчик:=значение downto конечное_значение do
```

```
 тело_цикла;
```

Счетчик – это переменная любого из перечисляемых типов (целого, булевого, символьного, диапазонного, перечисления). Начальные и конечные значения могут быть представлены не только значениями, но и выражениями, возвращающими совместимые с типом счетчика типы данных. Если между начальным и конечным выражением указано служебное слово **to**, то на каждом шаге цикла значение параметра будет увеличиваться на единицу. Если же указано **downto**, то значение параметра будет уменьшаться на единицу.

Количество итераций цикла **for** известно именно до его выполнения, но не до выполнения всей программы. Так в примере ниже, количество выполнений цикла определяется пользователем. Значение присваивается переменной, а затем используется в заголовке цикла. Но когда оно используется, циклу уже точно известно, сколько раз надо выполниться.

```
var
 i, n: integer;

begin
 write ('Количество знаков: ');
 readln (n);

 for i := 1 to n do
 write ('(*) ');

 readln
end.
```

### Перейти

Если число повторений заранее не известно и решение о завершении цикла принимается на основе анализа некоторого условия, то такой повторяющийся вычислительный процесс называется итерационным циклом

Цикл while

Цикл while является циклом с предусловием. В заголовке цикла находится логическое выражение. Если оно возвращает **true**, то тело цикла выполняется, если **false** – то нет.

Когда тело цикла было выполнено, то ход программы снова возвращается в заголовок цикла. Условие выполнения тела снова проверяется (находится значение логического выражения). Тело цикла выполнится столько раз, сколько раз логическое выражение вернет **true**. Поэтому очень важно в теле цикла предусмотреть изменение переменной, фигурирующей в заголовке цикла, таким образом, чтобы когда-нибудь обязательно наступала ситуация **false**. Иначе произойдет так называемое **зацикливание**, одна из самых неприятных ошибок в программировании.

```
var
 i, n: integer;
```

```

begin
 write ('Количество знаков: ');
 readln (n);

 i := 1;
 while i <= n do begin
 write ('(*) ');
 i := i + 1
 end;

 readln
end.

```

### Цикл repeat

Цикл **while** может не выполниться ни разу, если логическое выражение в заголовке сразу вернуло **false**. Однако такая ситуация не всегда может быть приемлемой. Бывает, что тело цикла должно выполниться хотя бы один раз, не зависимо оттого, что вернет логическое выражение. В таком случае используется цикл **repeat** – цикл с постусловием.

В цикле **repeat** логическое выражение стоит после тела цикла. Причем, в отличие от цикла **while**, здесь всё наоборот: в случае **true** происходит выход из цикла, в случае **false** – его повторение.

```

var
 i, n: integer;

begin
 write ('Количество знаков: ');
 readln (n);

 i := 1;
 repeat
 write ('(*) ');
 i := i + 1
 until i > n;

 readln
end.

```

В примере, даже если *n* будет равно 0, одна звездочка все равно будет напечатана.

**Разберем принцип работы циклов, называемых вложенными. Рассмотрим их на примере цикла с параметром:**

```

For <счетчик 1>:=<начальное знач.> to <конечное знач.> do <тело цикла 1>;
For <счетчик 2>:=<начальное знач.> to <конечное знач.> do <тело цикла 2>;
...
For <счетчик n>:=<начальное знач.> to <конечное знач.> do <тело цикла n>;

```

Вернувшись к терминологии, отметим, что в таких структурах одни циклы называются внешними, а другие внутренними. Так цикл 2 относительно цикла 1 является внутренним, а для цикла *n* он внешний.

Принцип работы вложенных циклов заключается в следующем. Сначала один раз выполняется цикл, идущий первым в программе, затем управление передается следующему циклу (№2), и если после него нет никаких других циклов, то он прокручивается до конца и снова единожды выполняется внешний цикл, иначе, если после цикла №2 стоит еще один вложенный цикл, то №2 выполняется один раз, передавая управление циклу, идущему после него. Такой алгоритм продолжается до тех пор, пока самый внешний цикл не выполнится необходимое количество раз.

Предположим у нас есть вложенные циклы **for**:

```
for i:=1 to m1 do
for j:=1 to m2 do
...
for k:=1 to mn do
```

Количество проходов по первому циклу =  $m_1$ , второму =  $m_2 * m_1$ , n-ому =  $m_n * m_{n-1} * m_{n-2} * \dots * m_2 * m_1$ . В качестве доказательства предлагаю разобраться в следующей программе.

```
1 program counter_performance;
2 uses crt;
3 var i, j, k, g, h: integer;
4 s_i, s_j, s_k, s_g, s_h: integer;
5 begin
6 s_i:=0; s_j:=0; s_k:=0; s_g:=0; s_h:=0;
7 for i:=1 to 5 do
8 begin
9 s_i:=s_i+1;
10 for j:=1 to 3 do
11 begin
12 s_j:=s_j+1;
13 for k:=1 to 19 do
14 begin
15 s_k:=s_k+1;
16 for g:=1 to 10 do
17 begin
18 s_g:=s_g+1;
19 for h:=1 to 6 do
20 s_h:=s_h+1;
21 end;
22 end;
23 end;
```

```

24 end;
25 write('i=',s_i,' j=',s_j,' k=',s_k,' g=',s_g,' h=',s_h);
26 readkey;
27 end.

```

Имеются 5 циклов со счётчиками от  $i$  до  $h$ . В каждом из них при помощи переменных подсчитывается, сколько раз выполняется каждый цикл. Запустив программу на своем компьютере, Вы должны увидеть такой результат:

```
i=5 j=15 k=285 g=2850 h=17100
```

Посчитав способом, изложенным выше, получите результат, соответствующий величинам, отображенным на скриншоте.

Вложенные конструкции аналогично работают в циклах с предусловием и постусловием. В следующей программе с помощью оператора **while** фигурка выводится на экран.

```

1 program nested_loops;
2 uses crt;
3 var i, j: integer;
4 begin
5 i:=1; j:=1;
6 while i<20 do begin
7 while j<i+1 do begin
8 write('\');
9 j:=j+1;
10 end;
11 j:=1;
12 i:=i+1;
13 writeln;
14 end;
15 readkey;
16 end.

```

При работе с массивами очень часто применяются вложенные циклы, а когда дело доходит до матриц (двумерных массивов), то даже их заполнение, как правило, делается через рассмотренные нами в этой статье конструкции.

### Двумерные и многомерные массивы

Представьте себе таблицу, состоящую из нескольких строк. Каждая строка состоит из

нескольких ячеек. Тогда для точного определения положения ячейки нам потребуется знать не одно число (как в случае таблицы линейной), а два: номер строки и номер столбца. Структура данных в языке Паскаль для хранения такой таблицы называется двумерным массивом. Описать такой массив можно двумя способами:

I.

*Var A : Array [1..20] Of Array [1..30] Of Integer;*

II.

*Var A : Array [1..20,1..30] Of Integer;*

В обоих случаях описан двумерный массив, соответствующий таблице, состоящей из 20 строк и 30 столбцов. Приведенные описания совершенно равноправны.

Отдельный элемент двумерного массива адресуется, естественно, двумя индексами. Например, ячейка, находящаяся в 5-й строке и 6-м столбце будет называться `A[5][6]` или `A[5,6]`.

### **Контрольные вопросы.**

1. Что такое цикл?
2. Какие циклы называются итерационными?
3. Как работает цикл **for**?
4. Какие виды цикла **for** существуют?
5. Объясните работу цикла с предусловием **while**
6. Объясните работу цикла с постусловием **repeat**
7. Объясните работу вложенных циклов

ДЗ. Изучить материал по конспекту. Решить задачу: написать программу заполнения одномерного массива минимальными элементами столбцов прямоугольного массива целых чисел