

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И СЕРВИСА

**РАБОЧАЯ ПРОГРАММА  
УЧЕБНОЙ ДИСЦИПЛИНЫ**

**ОП. 04 Основы алгоритмизации и программирования**

программы подготовки специалистов среднего звена  
09.02.06 Сетевое и системное администрирование

Форма обучения: очная

Владивосток 2021

Рабочая программа учебной дисциплины ОП.04 «Основы алгоритмизации и программирования» разработаны в соответствии с требованиями ФГОС СПО по специальности 09.02.06 Сетевое и системное администрирование, утвержденного приказом Минобрнауки РФ от 09.12.2016, №1548, примерной образовательной программой, рабочей программой учебной дисциплины.

Составитель:

Шуленина Алена Викторовна, преподаватель колледжа сервиса и дизайна ВГУЭС

Рассмотрена на заседании ЦМК направления Информационные системы и комплексы

Протокол № 9 от «12» 05 2021 г.

Председатель ЦМК  Е.А. Стефанович

## **СОДЕРЖАНИЕ**

- 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 3. УСЛОВИЯ РЕАЛИЗАЦИИ УЧЕБНОЙ ДИСЦИПЛИНЫ**
- 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ**

# 1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

## 1.1 Место дисциплины в структуре основной образовательной программы

Учебная дисциплина ОП.04 «Основы алгоритмизации и программирования» является частью общепрофессионального цикла основной образовательной программы (далее ООП) в соответствии с ФГОС СПО по специальности 09.02.06 Сетевое и системное администрирование.

## 1.2 Цель и планируемые результаты освоения дисциплины

По итогам освоения дисциплины, обучающиеся должны продемонстрировать результаты обучения, соотнесённые с результатами освоения ООП СПО, приведенные в таблице.

Код ПК, ОК	Умения	Знания
<i>ОК 01, ОК 02, ОК 04 ОК 05, ОК 09 ОК 10; ПК 1.2, ПК 2.3- ПК 2.4</i>	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. Выполнять проверку, отладку кода программы.	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, инкапсуляции и полиморфизма, наследования и переопределения.

## **2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

### **2.1 Объем учебной дисциплины и виды учебной работы**

<b>Вид учебной работы</b>	<b>Объем часов</b>
	73
в том числе:	
– теоретическое обучение	32
– практические занятия	32
– лабораторные занятия	не предусмотрено
– курсовая работа (проект)	не предусмотрено
– самостоятельная работа	3
– консультации	не предусмотрено
– промежуточная аттестация – экзамен	6

## 2.2. Тематический план и содержание учебной дисциплины

<i>Наименование разделов и тем</i>	<i>Содержание учебного материала и формы организации деятельности обучающихся</i>	<i>Объём в часах</i>	<i>Коды компетенций, формированию которых способствует элемент программы</i>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>Тема 1.</b> Основы алгоритмизации, языки и системы программирования.	<b><i>Содержание учебного материала</i></b>	<b>10</b>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ОК 10 ПК 1.2, ПК 2.3, ПК 2.4.
	Вводная лекция.		
	Основы алгоритмизации.	<b>6</b>	
	Алгоритмы цикла.		
	Основы алгоритмизации.		
	Языки и системы программирования		
	Алгоритмы. Языки программирования		
	<b><i>Тематика практических занятий и лабораторных работ</i></b>		
	Построение блок-схем	<b>1</b>	
<b><i>Самостоятельная работа обучающихся</i></b>	<b>1</b>		
Подготовка к защите практических работ			
<b>Тема 2.</b> Основные элементы языка. Управляющие операторы языка. Структурированные типы данных. Символьные типы данных	<b><i>Содержание учебного материала</i></b>	<b>40</b>	ОК 01 ОК 02 ОК 04 ОК 05 ОК 09 ОК 10 ПК 1.2, ПК 2.3, ПК 2.4.
	Основные элементы языка. Операторы языка. Ввод/вывод данных.	<b>20</b>	
	Управляющие операторы языка. Операторы выбора. Оператор условной передачи управления. Оператор безусловной передачи управления.		
	Оператор case. Операторы организации циклической обработки. Циклы.		
	Структуры данных. Массивы. Работа с массивами. Одномерные массивы. Обработка массивов. Сортировка массивов. Двумерные массивы. Решение систем уравнений.		
	Коллекции. Контейнеры. Операции над коллекциями и контейнерами. Обработка коллекций. Многомерные контейнеры. Обработка контейнеров.		
	Символьные типы данных. Символы и строки. Обработка символов. Обработка строк.		
	Строковые массивы. Файлы. Поток. Считывание из файла. Запись в файл. Редактирование файлов.		
	<b><i>Тематика практических занятий и лабораторных работ</i></b>		
	Операторы выбора, циклы		
	Работа с массивами		
	Коллекции и контейнеры		
	Символы и строки		
	Работа с файлами		
	<b><i>Самостоятельная работа обучающихся</i></b>	<b>1</b>	

	Подготовка к защите практических работ		
<b>Тема 3.</b> Модульное программирование. Рекурсия. Визуально-событийно управляемое программирование. Разработка оконного приложения	<b><i>Содержание учебного материала</i></b>	<b>14</b>	ОК 01
	Локальные и глобальные переменные. Модульное программирование.	<b>6</b>	ОК 02
	Процедуры и функции. Подпрограммы. Передача данных в процедуры и функции.		ОК 04
	Рекурсия. Разработка рекурсивных подпрограмм		ОК 05
	Визуально-событийно управляемое программирование.		ОК 09
	Виджеты. События. Основные элементы управления.		ОК 10
	Разработка оконного приложения.		ПК 1.2, ПК 2.3, ПК 2.4.
	Установка приложения	<b>8</b>	
	<b><i>Тематика практических занятий и лабораторных работ</i></b>		
	Реализация подпрограммы. Разработка рекурсивных подпрограмм		
	Основные элементы управления. Разработка оконного приложения		
	<b><i>Самостоятельная работа обучающихся</i></b>	<b>1</b>	
	Подготовка к защите практических работ		
<b>Всего:</b>		<b>67</b>	

### **3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»**

#### **3.1 Материально-техническое обеспечение**

Для реализации программы учебной дисциплины предусмотрены следующие специальные помещения:

**Лаборатория программного обеспечения компьютерных сетей, программирования и баз данных:**

количество посадочных мест – 25, стол для преподавателя 1 шт., стул для преподавателя 1 шт., шкаф 3 шт., компьютерный стол 20 шт., персональный компьютер ПК i3 2120/500Gb/4Gb 20 шт., мультимедийный комплект: проектор, интерактивная доска Elite Panaboard UBT-T880W 1 шт., колонки ОКЛИК 1 шт., доска маркерная меловая комбинированная 1 шт., информационный стенд 2 шт., сервер (процессор-i7-6700 (4 ядра, 3.4Ghz, L3 8 Mb), оперативная память-32Gb; накопитель-HDD 5 Tb) 1 шт.,

типовой состав для монтажа и наладки компьютерной сети: набор для обжима и тестирования кабеля UTP, кабель UTP - 305м 1шт., коннекторы 8P8C, 12 шт., коммутатор: CISCO WS-C2960-24TT-L 1 шт., блок бесперебойного питания, фильтр: APC 2200 1 шт., фильтр 6 розеток 6 шт., дидактические пособия.

ПО: 1. Windows 7(профессиональная лицензия, ООО "Битроникс Владивосток"

Контракт№ 0320100030814000018-45081 от 09.09.14 № 48609744, №62096196, № 48958910, № 45829305, бессрочно);

2. MS Office 2010 pro (лицензия № 48958910, № 47774898 , бессрочно);

3. visual c++ 2008 express edition (свободное),

4. oracle vm virtualbox (свободное),

5. cisco packet tracer (свободное),

6. micosoft SQL server 2008 (свободное),

7. k-lite codec pack (свободное),

8. visual studio 2008 (свободное),

9. Google Chrome (свободное);

10. Internet Explorer (свободное)

#### **3.2. Информационное обеспечение реализации программы**

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

##### **Основная литература**

1. Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — М. : ФОРУМ : ИНФРА-М, 2020. — 431 с. — Текст: электронный. - URL: <https://new.znaniy.com/catalog/product/1072040> (дата обращения: 03.03.2020)

2. Колдаев, В. Д. Основы алгоритмизации и программирования : учеб. пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-16-103967-0. - Текст : электронный. - URL: <https://new.znaniy.com/catalog/product/980416> (дата обращения: 03.03.2020)

3. Фризен, И. Г. Основы алгоритмизации и программирования (среда PascalABC.NET) : учебное пособие / И.Г. Фризен. — М.: ФОРУМ : ИНФРА-М, 2020. — 392 с. — Текст :

электронный. - URL: <https://new.znaniium.com/catalog/product/1047096> (дата обращения: 03.03.2020)

#### Дополнительная литература

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учеб. пособие / С.Р. Гуриков. — М.: ФОРУМ : ИНФРА-М, 2020. — 343 с. — Текст : электронный. - URL: <https://new.znaniium.com/catalog/product/1042452> (дата обращения: 03.03.2020)

2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке С# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — М. : Юрайт, 2019. — 322 с. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/431505> (дата обращения: 03.03.2020).

3. Трофимов, В. В. Основы алгоритмизации и программирования : учебник для среднего профессионального образования / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — М.: Юрайт, 2020. — 137 с. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454452> (дата обращения: 03.03.2020).

#### 4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ «ОП. 04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

<i>Результаты обучения</i>	<i>Критерии оценки</i>	<i>Формы и методы оценки</i>
<p><i>Перечень знаний, осваиваемых в рамках дисциплины:</i></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 и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.</p>	<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</p>	<p>Оценка в рамках текущего контроля результатов выполнения индивидуальных контрольных заданий, результатов выполнения практических работ, устный индивидуальный опрос.</p> <p>Письменный опрос в форме тестирования</p>

<p><i>Перечень умений, осваиваемых в рамках дисциплины:</i></p>	<p>с освоенным материалом в основном сформированы, большинство</p>	
<p>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. Выполнять проверку, отладку кода программы.</p>	<p>предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.  «Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.</p>	<p>Экспертное наблюдение и оценивание выполнения практических работ.  Текущий контроль в форме защиты практических работ</p>

МИНОБРНАУКИ РОССИИ  
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И СЕРВИСА

**КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА**  
для проведения текущего контроля и промежуточной аттестации  
по учебной дисциплине

**ОП.04 Основы алгоритмизации и программирования**

программы подготовки специалистов среднего звена  
09.02.06 Сетевое и системное администрирование

Форма обучения: очная

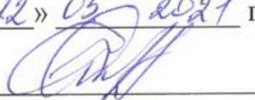
Владивосток 2021

Контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации по учебной дисциплине ОП.04 «Основы алгоритмизации и программирования» разработаны в соответствии с требованиями ФГОС СПО по специальности 09.02.06 Сетевое и системное администрирование, утвержденного приказом Минобрнауки РФ от 09.12.2016, №1548, примерной образовательной программой, рабочей программой учебной дисциплины.

Разработчик: А.В. Шуленина, преподаватель колледжа сервиса и дизайна ВГУЭС

Рассмотрена на заседании ЦМК направления Информационные системы и комплексы

Протокол № 9 от «12» 05 2021 г.

Председатель ЦМК  Е.А. Стефанович

## 1 Общие положения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Основы алгоритмизации и программирования.

КОС включают в себя контрольные материалы для проведения текущего контроля успеваемости и промежуточной аттестации по дисциплине, которая проводится в форме дифференцированного зачёта.

## 2 Планируемые результаты обучения по дисциплине, обеспечивающие результаты освоения образовательной программы

В результате освоения дисциплины студент 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.
- Выполнять проверку, отладку кода программы.

В результате освоения дисциплины студент 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, инкапсуляции и полиморфизма, наследования и переопределения.

### Перечень основных показателей оценки результатов, элементов знаний и умений, подлежащих текущему контролю и промежуточной аттестации

Результаты обучения (ПК)	Код и наименование элемента умений	Код и наименование элемента знаний
ПК 1.2 Осуществлять выбор технологий, инструментальных средств и средств вычислительной техники при организации процесса разработки и исследования объектов профессиональной деятельности	У1	31
	У2	32
	У4	33
	У5	
ПК 2.3 Обеспечивать сбор данных для анализа использования и функционирования программно-технических средств компьютерных сетей	У3	33-34

ПК 2.4 Взаимодействовать со специалистами смежного профиля при разработке методов, средств и технологий применения объектов профессиональной деятельности.	У4 У6 У7	35
--	----------------	----

### 3 Распределение оценивания результатов обучения по видам контроля

Код и наименование элемента умений или знаний	Виды аттестации	
	Текущий контроль	Промежуточная аттестация
У1 Разрабатывать алгоритмы для конкретных задач.	Оценка выполнения практического задания	Решение расчетных заданий дифференцированного зачета
У2 Использовать программы для графического отображения алгоритмов.	Оценка выполнения практического задания	
У3 Определять сложность работы алгоритмов.	Оценка выполнения практического задания	
У4 Работать в среде программирования.	Оценка выполнения практического задания	
У5 Реализовывать построенные алгоритмы в виде программ на конкретном языке программирования.	Оценка выполнения практического задания	
У6 Оформлять код программы в соответствии со стандартом кодирования.	Оценка выполнения практического задания	
У7 Выполнять проверку, отладку кода программы.	Оценка выполнения практического задания	
31 Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции.	Оценка по результатам собеседования	
32 Эволюцию языков программирования, их классификацию, понятие системы программирования.	Оценка по результатам собеседования	
33 Основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти.	Оценка по результатам собеседования	
34 Подпрограммы, составление библиотек подпрограмм.	Оценка выполнения практического задания	
35 Объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.	Оценка по результатам собеседования	

### 3.1. Формы текущего контроля по учебной дисциплине в ходе освоения

Элементы учебной дисциплины	Формы текущего контроля
<b>Раздел 1. Основы алгоритмизации, языки и системы программирования.</b>	
Тема 1.1. Вводная лекция.	
Тема 1.2. Основы алгоритмизации.	
Тема 1.3. Алгоритмы цикла.	Практическое занятие №1
Тема 1.4. Основы алгоритмизации.	
Тема 1.5. Языки и системы программирования	
Тема 1.6. Алгоритмы. Языки программирования	
<b>Раздел 2. Основные элементы языка. Управляющие операторы языка. Структурированные типы данных. Символьные типы данных.</b>	
Тема 2.1. Основные элементы языка. Операторы языка. Ввод/вывод данных.	
Тема 2.2. Управляющие операторы языка. Операторы выбора. Оператор условной передачи управления. Оператор безусловной передачи управления.	Практическое занятие №2
Тема 2.3. Оператор case. Операторы организации циклической обработки. Циклы.	Практическое занятие №3
Тема 2.4. Структуры данных. Массивы. Работа с массивами. Одномерные массивы. Обработка массивов. Сортировка массивов. Двумерные массивы. Решение систем уравнений.	Практическое занятие №4
Тема 2.5. Коллекции. Контейнеры. Операции над коллекциями и контейнерами. Обработка коллекций. Многомерные контейнеры. Обработка контейнеров.	
Тема 2.6. Символьные типы данных. Символы и строки. Обработка символов. Обработка строк.	
Тема 2.7. Строковые массивы. Файлы. Потоки. Считывание из файла. Запись в файл. Редактирование файлов.	
<b>Раздел 3. Модульное программирование. Рекурсия. Визуально-событийно управляемое программирование. Разработка оконного приложения.</b>	
Тема 3.1. Разработка оконного приложения	
Тема 3.2. Локальные и глобальные переменные. Модульное программирование.	
Тема 3.3. Процедуры и функции. Подпрограммы. Передача данных в процедуры и функции.	Практическое занятие №5
Тема 3.4. Рекурсия. Разработка рекурсивных подпрограмм	
Тема 3.5. Визуально-событийно управляемое программирование.	Практическое занятие №6
Тема 3.6. Виджеты. События. Основные элементы управления.	
Тема 3.7. Разработка оконного приложения.	Практическое занятие №7,8
Тема 3.8. Установка приложения	
	<b>Дифференцированный зачет</b>

#### 4 Перечень оценочных средств

№ п/п	Форма проведения оценки результатов освоения дисциплины	Краткая характеристика форм оценки результатов освоения дисциплины	Представление оценочного средства в фонде
1.	Расчетно-графическая работа	Средство проверки умений применять полученные знания по заранее определенной методике для решения задач или заданий по модулю или дисциплине в целом.	Комплект заданий для выполнения расчетно-графической работы
2.	Собеседование	Средство контроля, организованное как специальная беседа преподавателя с обучающимся на темы, связанные с изучаемой дисциплиной, и рассчитанное на выяснение объема знаний обучающегося по определенному разделу, теме, проблеме и т.п.	Вопросы по темам/разделам дисциплины
3.	Творческое задание	Частично регламентированное задание, имеющее нестандартное решение и позволяющее диагностировать умения, интегрировать знания различных областей, аргументировать собственную точку зрения. Может выполняться в индивидуальном порядке или группой обучающихся.	Темы групповых и/или индивидуальных творческих заданий

#### 5 Контрольно-измерительные материалы для текущего контроля по учебной дисциплине

Практическое занятие №1. Составление блок-схем алгоритмов.

**Контрольная работа по темам 1.2.–1.3.**

**Вариант №1**

**№1** Какие значения будут присвоены переменным **x** и **i** после выполнения следующих фрагментов программ?

```
i: =n;
while i<5 do
begin
    x: =x+1;
    i: =i+1;
end;
при 1) n=6; x=0
     2) n=4; x=0
```

**№2.** Укажите различия циклов **WHILE** и **REPEAT**.

**Вариант №2**

**№1** Какие значения будут присвоены переменным **i** и **s** после выполнения следующих фрагментов программ?

```
i: =n
s: =0
while (i<>4) or (y='нет') do
```

begin

```
i: =i+2;  
s: =s+1;  
end;
```

при 1) n=1; y='нет'  
2) n=5; y='да'

№2 Когда нельзя применять цикл **FOR**?

### **Вариант №3**

№1 Какие значения будут присвоены переменным **x** и **i** после выполнения следующих фрагментов программ?

```
i: =n;  
x: =0;  
while (i<n) and (y='да') do  
  begin  
    x: =x+1;  
    i: =i-1
```

end

при 1) n=3; y='так'  
2) n=1; y='нет'

№2. Что такое переменная счетчик цикла? Какого типа она может быть в арифметическом и логическом циклах?

### **Вариант №4**

№1. Какие значения будут присвоены переменным **x** и **i** после выполнения следующих фрагментов программ?

```
n: =1;  
x: =0  
repeat  
  x: =x+1  
  i: = i+1  
until (1<n) and (y<>'так')
```

при: 1) n=3; y='да'  
2) n=1; y='нет'

№2. При каких обстоятельствах программа может «зациклиться»?

Практическое занятие №2. Составление программ разветвляющейся структуры.

Практическое занятие №3. Составление программ циклической структуры.

Практическое занятие №4. Обработка одномерных и двумерных массивов.

Практическое занятие №5. Организация и использование процедур и функций.

Практическое занятие №6. Объявление класса, создание экземпляров класса и наследованного класса.

Практическое занятие №7. Разработка оконного приложения.

Практическое занятие №8. Разработка многооконного приложения

## **КРИТЕРИИ ОЦЕНКИ ЗНАНИЙ И УМЕНИЙ**

Оценка «отлично» выставляется, если студент правильно ответил теоретический вопрос и в качестве результата выполнения практического задания предъявил работающую программу, причем программа реализует некоторый интерфейс пользователя.

Оценка «хорошо» выставляется, если студент в качестве результата выполнения практического задания предъявил работающую программу (причем программа не реализует интерфейс пользователя), а ответил на теоретический вопрос с небольшими ошибками.

Оценка «удовлетворительно» выставляется, если студент ответил на теоретический вопрос (с небольшими ошибками), а в качестве результата выполнения практического задания предъявил программу, решающую его большую часть.

Оценка «неудовлетворительно» выставляется, если студент не выполняет практическое задание.

## 5.2 Контрольно-измерительные материалы для промежуточной аттестации по учебной дисциплине

### Задания для проведения дифференцированного зачета

<b>Вопросы и задания</b>
<b>Теоретические вопросы</b>
Понятие алгоритма. Основные свойства алгоритма.
Словесное описание алгоритма. Графическое изображение алгоритма. Правила составления блок-схем алгоритмов.
Понятие линейного алгоритма. Примеры линейных алгоритмов.
Разновидности условного алгоритма(полное разветвление, неполное разветвление, схема множественного выбора). Примеры условных алгоритмов.
Виды циклических алгоритмов (цикл ПОКА, цикл ДО, цикл с параметрами). Примеры использования циклов.
Понятие функционального алгоритма. Примеры составления подалгоритмов.
Понятие комбинированных алгоритмов. Основные алгоритмы обработки массивов. Примеры использования.
Методы алгоритмизации. Этапы решения задач на ПК.
Синтаксис языка СИ (алфавит, идентификаторы, комментарии, разделители).
Понятие препроцессора.
Типы данных языка СИ. Объявления.
Понятие переменных. Объявление переменных. Инициализация переменных.
Понятие констант. Объявление констант.
Функции форматированного ввода/вывода данных scanf(), printf(). Примеры использования.
Операции языка СИ (арифметические, отношения, логические, побитовые, условия, запятая).
Условные операторы if, if-else ,оператор безусловного перехода goto. Примеры использования.
Оператор множественного выбора switch. Примеры использования.
Оператор цикла for. Примеры использования.
Оператор цикла while. Примеры использования.
Оператор цикла do-while. Примеры использования.
Понятие массива. Размерность массива. Обращение к элементам массива.
Одномерные массивы. Двумерные массивы. Инициализация массивов. Примеры обработки массивов.
Понятие указателя, операции над указателями. Связь указателей с массивами. Примеры использования.
Массивы указателей, инициализация указателей. Примеры обработки массивов с помощью указателей.
Строки как массивы символов. Функции работы со строками встроенной библиотеки языка. Примеры программ обработки строк.

<b>Вопросы и задания</b>
Понятие пользовательских типов данных. Объявление структуры. Примеры программ обработки массива структур.
Понятие функции, Объявление и определение функций.
Параметры функции. Механизмы передачи параметров. Передача имен функций в качестве параметров. Примеры программ с использованием функций.
Глобальные и локальные переменные. Время жизни переменных и область видимости. Классы памяти.
Директивы препроцессора. Заголовочные файлы и файлы реализации.
Понятие файлового потока. Основные принципы управления файловыми потоками. Указатель на файл.
Произвольная запись в файл. Произвольное чтение из файла. Примеры программ работы с файлами. Примеры программ работы с файлами.
Функции работы с файлами. Последовательная запись в файл. Последовательное чтение из файла. Примеры программ работы с файлами.
Основные понятия объектно-ориентированного программирования: объекты и классы.
Объявление класса. Методы класса. Создание методов. Конструктор и деструктор класса.
Дружественные функции и классы. Доступ к компонентам класса. Объекты класса.
Создание объекта. Конструктор объекта. Присваивание и инициализация объектов. Передача объектов функциям. Массивы объектов.
<b>Типовые практические задания</b>
Вычислить значение выражения $y = \frac{x^n}{(n+1)} * 1/4$ ; где x, n – задаются пользователем с клавиатуры. Результаты вывести на экран.
Определить сколько положительных, четных значений попадает в промежуток от -12 до +12. Результат вывести на экран.
Организовать диалог пользователя и компьютера по заполнению паспортных данных.
Составить программу вычислений следующих функций: $y = \cos \frac{(2x)}{6} - \frac{\sin(4x)}{8}$ ; $f = \frac{x^2+ax+b}{c}$ ; $z = \frac{xy-0.5f}{5}$ ; Результаты вывести на экран.
Определить, какое из трех значений, вводимых пользователем с клавиатуры, является кратным 5. Вывести на экран соответствующее сообщение.
Определить количество отрицательных значений из 15 чисел, вводимых пользователем с клавиатуры. Результат вывести на экран.
Определить разность всех четных чисел, вводимых пользователем с клавиатуры. Ввод данных производить до тех пор, пока не встретится число 0. Результаты вывести на экран.
Вычислить сумму: $S = \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \dots + \frac{1}{2^n}$ ; где n изменяется от 1 до 10. Результат вывести на экран.
Вычислить $y = x^{10} + x^8 + x^6 + \dots + x^0(x^n)$ ; где n изменяется от 10 до 0 с шагом 2. Значение x вводится пользователем с клавиатуры. Результаты вывести на экран.
Вычислить значение выражения $y = \frac{x^2+n^2}{(2*n+1)} * 1/3$ ; где x, n – задаются пользователем с клавиатуры. Результаты вывести на экран.
Определить сколько положительных значений, кратных 3 попадает в промежуток от -15 до +15. Результат вывести на экран.
Найти среднеарифметическое значение чисел от 10 до 25. Результат вывести на экран.
Составить программу вычислений следующих функций: $y = \sin \frac{3x}{3} - \frac{\cos(4x)}{4}$ ; $f = \frac{x^4+ax^2+bx}{c}$ ; $z = \frac{5y+3f}{x}$ ; Результаты вывести на экран.
Заполнить одномерный целочисленный массив, размером N, числами. Заменить все числа кратные 8 на 0. Вывести массив на экран до и после модификации.

<b>Вопросы и задания</b>
Заполнить двумерный целочисленный массив, размером N, числами. Зеркально отобразить его содержимое по горизонтали. Вывести массив на экран до и после модификации.
Заполнить двумерный массив случайными числами. Зеркально отобразить его содержимое по вертикали. Вывести массив на экран до и после модификации.
Заполнить двумерный массив случайными числами. Повернуть его содержимое на 90 градусов. Вывести массив на экран до и после модификации.
Заполнить двумерный массив случайными числами. Повернуть его содержимое на 180 градусов. Вывести массив на экран до и после модификации.
Заполнить одномерный массив случайными числами в диапазоне от 0 до 50. Определить среднее арифметическое всех чисел входящих в массив и записать это значение в последний элемент массива. Вывести массив на экран до и после модификации.
Заполнить одномерный массив A размерностью 2n случайными числами. Создать одномерный массив C размерностью n и заполнить его элементами массива A по формуле $C[i]=A[i]*A[i+n+1]$ . Вывести массивы A и C на экран.
Заполнить одномерный массив случайными числами. Поменять местами максимальный и минимальный элементы массива. Вывести массив на экран до и после модификации.
Заполнить одномерный массив случайными числами. Посчитать количество элементов между максимальным и минимальным элементами массива. Вывести массив на экран до и после модификации.
Заполнить два одномерных массива случайными числами. Вывести на экран оба массива и числа присутствующие в обоих массивах одновременно.
Заполнить одномерный массив размерностью $> 30$ элементами случайными целыми числами из диапазона от 10 до 20. Создать второй массив и перенести в него элементы первого массива, но так, чтобы они встречались в нем только один раз. Вывести оба массива на экран.
Заполнить одномерный массив случайными числами. Используя побитовые операции прибавить ко всем четным числам 1. Вывести массив на экран до и после модификации.
Заполнить одномерный массив случайными числами. Используя побитовые операции прибавить ко всем нечетным числам 1. Вывести массив на экран до и после модификации.
Заполнить одномерный массив вещественными числами по формуле , где x номер элемента массива. Вывести массив на экран.
Заполнить одномерный массив вещественными числами по формуле , где A, B, C день и месяц сдачи экзамена и номер билета соответственно, а x номер элемента массива. Вывести массив на экран.
Заполнить одномерный массив размерностью $> 30$ элементами случайными целыми числами из диапазона от 10 до 20. Посчитать какое количество элементов встречается в массиве больше 2 раз. Вывести массив и результат на экран.
Вывести на экран таблицу умножения.
Дан двумерный целочисленный массив A размером 4x4. Заполнить массив числами. Определить максимальный элемент массива. Вывести на экран полученный массив и значение максимального элемента.
Дан двумерный целочисленный массив A размером 4x4. Заполнить массив числами. Определить минимальный элемент массива. Вывести на экран полученный массив и значение минимального элемента.
Написать программу, которая запрашивает две строки, а затем соединяет их и выводит новую строку на экран.
Написать программу, которая запрашивает десятичное число и вычисляет сумму цифр четных позиций.
Написать программу, которая запрашивает десятичное число и вычисляет сумму цифр нечетных позиций.

**Вопросы и задания**

Написать программу, которая вычисляет сумму цифр натурального числа, введенного с клавиатуры.

Дана целочисленная прямоугольная матрица. Вычислить количество строк, содержащих хотя бы один нулевой элемент.

Дана целочисленная прямоугольная матрица. Вычислить количество положительных элементов по каждому столбцу.

Написать программу, которая считывает текст из файла и выводит на экран количество восклицательных предложений.

Составить функцию определения простого числа. Используя функцию определить количество простых чисел в массиве.

Построить описание класса содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменения размеров.

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

Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости.

Составить описание класса для представления комплексных чисел с возможностью задания вещественной и мнимой частей как числами типов `double`, так и целыми числами.

Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами.