

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**РАБОЧАЯ ПРОГРАММА
УЧЕБНОЙ ДИСЦИПЛИНЫ**

ОП.06 Основы алгоритмизации и программирования
программы подготовки специалистов среднего звена
09.02.01 Компьютерные системы и комплексы

Форма обучения: *очная*

Рабочая программа учебной дисциплины ОП.06 «Основы алгоритмизации и программирования» разработана в соответствии с требованиями Федерального государственного образовательного стандарта среднего профессионального образования по специальности 09.02.01 Компьютерные системы и комплексы, утверждённого приказом Минпросвещения России от 25 мая 2022 г. № 362, примерной образовательной программой.

Разработчик: Р.С. Реуцкий, преподаватель

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 9 от « 18 » 05 2026 г.

Председатель ЦМК  Е.А. Стефанович
подпись

СОДЕРЖАНИЕ

- 1 ОБЩАЯ ХАРАКТЕРИСТИКА РАБОЧЕЙ 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.06 Основы алгоритмизации и программирования является частью общепрофессионального учебного цикла основной образовательной программы (далее ООП) в соответствии с ФГОС СПО по специальности 09.02.01 Компьютерные системы и комплексы.

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 компетенции	Умения	Знания
ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2	Разрабатывать и анализировать алгоритмы для решения поставленных задач; определять сложность алгоритмов; реализовывать типовые алгоритмы в виде программ на актуальных языках программирования; использовать средства проектирования для создания и графического отображения алгоритмов; оформлять код программ в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы	Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; классификация языков программирования; понятие системы программирования; основные элементы языка, структура программы; методы реализации типовых алгоритмов; операторы и операции, управляющие структуры, структуры данных, классы памяти; понятие подпрограммы, библиотеки подпрограмм; объектно-ориентированная 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

2.1 Объем учебной дисциплины и виды учебной работы

Вид учебной работы	Объем часов
Объем образовательной программы учебной дисциплины	104
в том числе:	
– теоретическое обучение	32
– практические занятия (если предусмотрено)	50
– самостоятельная работа	12
– консультация	10
– промежуточная аттестация – дифференцированный зачет	

2.2 Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах	Коды компетенций, формированию которых способствует элемент программы	
1	2	3	4	
Тема 1. Основы алгоритмизации, языки и системы программирования	Содержание учебного материала		ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2	
	1. Основы алгоритмизации. Алгоритмы цикла.	6		
	2. Системы программирования. Языки программирования			
	Практическое занятие № 1. Построение блок-схем	6		
	Самостоятельная работа обучающихся Обучающемуся необходимо разработать алгоритм решения задачи (например, нахождение суммы чисел от 1 до N или определение простоты числа) и представить его в виде блок-схемы. Обучающемуся необходимо выполнить сравнительный анализ языков программирования (компилируемые и интерпретируемые) с приведением примеров и выводом.	4		
Тема 2. Основные элементы языка. Управляющие операторы языка. Структурированные типы данных. Символьные типы данных	Содержание учебного материала		ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2	
	1. Основные элементы языка. Операторы языка. Ввод/вывод данных.	20		
	2. Управляющие операторы языка. Операторы выбора. Оператор условной передачи управления. Оператор безусловной передачи управления.			
	3. 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. Потоки. Считывание из файла. Запись в файл. Редактирование файлов.			
	Практическое занятие № 2. Операторы выбора, циклы			6
	Практическое занятие № 3. Работа с массивами			6

	Практическое занятие № 4. Коллекции и контейнеры	6	
	Практическое занятие № 5. Символы и строки	6	
	Практическое занятие № 6. Работа с файлами	6	
	Самостоятельная работа обучающихся Обучающемуся необходимо реализовать программу, выполняющую обработку числовых данных, например, формирование таблицы умножения для заданного числа. Обучающемуся необходимо реализовать программу работы с массивом, включающую заполнение, сортировку и вывод результатов.	4	
Тема 3. Модульное программирование. Рекурсия. Визуально-событийно управляемое программирование. Разработка оконного приложения	Содержание учебного материала		ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2
	1. Локальные и глобальные переменные. Модульное программирование.	6	
	2. Процедуры и функции. Подпрограммы. Передача данных в процедуры и функции.		
	3. Рекурсия. Разработка рекурсивных подпрограмм		
	4. Визуально-событийно управляемое программирование.		
	5. Виджеты. События. Основные элементы управления.		
	6. Разработка оконного приложения.		
	7. Установка приложения		
	Практическое занятие № 7. Реализация подпрограммы. Разработка рекурсивных подпрограмм	6	
	Практическое занятие № 8. Основные элементы управления. Разработка оконного приложения	8	
Самостоятельная работа обучающихся Обучающемуся необходимо реализовать программу с использованием процедур и функций для решения вычислительной задачи. Обучающемуся необходимо разработать простое оконное приложение с использованием элементов управления и обработкой пользовательских событий.	4		
Консультация		10	
Промежуточная аттестация дифференцированный зачет			
Всего:		104	

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ

3.1 Материально-техническое обеспечение

Для реализации программы учебной дисциплины предусмотрено следующие специальные помещения:

Кабинет информационных технологий:

количество посадочных мест – 25, стол для преподавателя 1 шт., стул для преподавателя 1 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., звуковые колонки 1 шт., доска маркерная меловая комбинированная 1 шт., информационный стенд 2 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. Autodesk AutoCAD 2019 Edu (свободное);
4. visual c++ 2008 express edition (свободное),
5. oracle vm virtualbox (свободное),
6. cisco packet tracer (свободное),
7. microsoft SQL server 2008 (свободное),
8. k-lite codec pack (свободное),
9. visual studio 2008 (свободное).

Мастерская по компетенции «Облачные технологии»

Персональный компьютер Lenovo ThinkCentreTiny M75 + TiO 24 – 21 шт;
Мультимедийный комплект;

Многофункциональное устройство №2 МФУ Xerox WC3345;

Маршрутизатор №3 MikroTik CCR1016-12G MikroTik.

программное и методическое обеспечение

1. Office Professional Plus 2019 Russian OLV NL Each AcademicEdition Additional Product Microsoft Ireland Operations Limited;
2. Microsoft Windows 10.
3. Oracle vm virtualbox
4. VMware Workstation

3.2 Информационное обеспечение реализации программы

Для реализации программы учебной дисциплины библиотечный фонд ВВГУ укомплектован печатными и электронными изданиями.

Обучающиеся из числа инвалидов и лиц с ограниченными возможностями здоровья обеспечены печатными и (или) электронными образовательными ресурсами в формах, адаптированных к ограничениям их здоровья.

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 : ИНФРА-М, 2026. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-16-021353-8. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2222607> (дата обращения: 01.04.2026).

2. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ИНФРА-М, 2026. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-16-021186-2. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2207916> (дата обращения: 01.04.2026).

3. 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.ru/catalog/product/1902735> (дата обращения: 01.04.2026).

Дополнительная литература

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2026. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-16-016906-4. - Текст : электронный. - URL: <https://znanium.ru/catalog/product/2216924> (дата обращения: 01.04.2026).

2. Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для среднего профессионального образования / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2019. — 322 с. — (Профессиональное образование). — ISBN 978-5-534-10772-2. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/431505> (дата обращения: 01.04.2026).

4 КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ УЧЕБНОЙ ДИСЦИПЛИНЫ

Результаты обучения	Критерии оценки	Формы и методы оценки
Перечень знаний, осваиваемых в рамках дисциплины: Понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции; классификация языков программирования; понятие системы программирования; основные элементы языка, структура программы; методы реализации типовых алгоритмов; операторы и операции, управляющие структуры, структуры данных, классы памяти; понятие подпрограммы, библиотеки подпрограмм; объектно-ориентированная модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического	«Отлично» - теоретическое содержание курса освоено полностью, без пробелов, умения сформированы, все предусмотренные программой учебные задания выполнены, качество их выполнения оценено высоко. «Хорошо» - теоретическое содержание курса освоено полностью, без пробелов, некоторые умения сформированы недостаточно, все предусмотренные программой учебные задания выполнены, некоторые виды заданий выполнены с ошибками. «Удовлетворительно» - теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые умения работы с освоенным	Оценка в рамках текущего контроля результатов выполнения индивидуальных контрольных заданий, результатов выполнения практических работ, устный индивидуальный опрос. Письменный опрос в форме тестирования

языка: понятие классов и объектов, их свойств и методов, инкапсуляции и полиморфизма, наследования и переопределения.	материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые из выполненных заданий содержат ошибки.	
Перечень умений, осваиваемых в рамках дисциплины: Разрабатывать и анализировать алгоритмы для решения поставленных задач; определять сложность алгоритмов; реализовывать типовые алгоритмы в виде программ на актуальных языках программирования; использовать средства проектирования для создания и графического отображения алгоритмов; оформлять код программ в соответствии со стандартом кодирования; выполнять проверку, отладку кода программы	«Неудовлетворительно» - теоретическое содержание курса не освоено, необходимые умения не сформированы, выполненные учебные задания содержат грубые ошибки.	Экспертное наблюдение и оценивание выполнения практических работ. Текущий контроль в форме защиты практических работ

Для оценки достижения запланированных результатов обучения по дисциплине разработаны контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации, которые прилагаются к рабочей программе дисциплины.

МИНОБРНАУКИ РОССИИ
ВЛАДИВОСТОКСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ

КОНТРОЛЬНО-ОЦЕНОЧНЫЕ СРЕДСТВА
для проведения текущего контроля и промежуточной аттестации
учебной дисциплины

ОП.06 Основы алгоритмизации и программирования
программы подготовки специалистов среднего звена

09.02.01 Компьютерные системы и комплексы

Форма обучения: очная

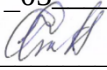
Владивосток 2026

Контрольно-оценочные средства для проведения текущего контроля и промежуточной аттестации профессионального модуля ОП.06 «Основы алгоритмизации и программирования» разработаны в соответствии с требованиями ФГОС СПО по специальности 09.02.01 Компьютерные системы и комплексы, утверждённого приказом Минпросвещения России от 25 мая 2022 г. № 362, примерной образовательной программой, рабочей программой учебной дисциплины.

Разработчик(и): Р.С. Реуцкий, преподаватель

Рассмотрено и одобрено на заседании цикловой методической комиссии

Протокол № 9 от «_18_» _05_ 2026 г.

Председатель ЦМК  Е.А. Стефанович
подпись

1 Общие сведения

Контрольно-оценочные средства (далее – КОС) предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу профессионального модуля ОП.06 «Основы алгоритмизации и программирования».

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

2 Планируемые результаты обучения по профессиональному модулю, обеспечивающие результаты освоения образовательной программы

Код ОК, ПК	Код результата обучения ¹	Наименование результата обучения ¹
ОК 01 ОК 02 ПК 1.1 ПК 2.1 ПК 2.2	У1	разрабатывать и анализировать алгоритмы для решения поставленных задач
	У2	определять сложность алгоритмов
	У3	реализовывать типовые алгоритмы в виде программ на актуальных языках программирования
	У4	использовать средства проектирования для создания и графического отображения алгоритмов
	У5	оформлять код программ в соответствии со стандартом кодирования
	У6	выполнять проверку, отладку кода программы
	31	понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции
	32	классификация языков программирования
	33	понятие системы программирования
	34	основные элементы языка, структура программы
	35	методы реализации типовых алгоритмов
	36	понятие подпрограммы, библиотеки подпрограмм
	37	объектно-ориентированная 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

3 Соответствие оценочных средств контролируемым результатам обучения

3.1 Средства, применяемые для оценки уровня теоретической подготовки

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
Раздел 1. Вычислительные приборы и устройства				
Тема 1. Основы алгоритмизации, языки и системы программирования	У1	Разрабатывает и анализирует алгоритмы для решения задач, корректно выбирает алгоритмическую структуру	Устный опрос (вопросы 1–6, 9–10) (5.1); Тест №1 (5.3); Рефераты	Экзаменационные билеты №1–2 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
			(темы 1–5, 7–9) (5.2)	
	У2	Определяет сложность алгоритмов и обосновывает выбор алгоритмического решения	Устный опрос (вопросы 2, 3, 30) (5.1); Рефераты (темы 10) (5.2)	Экзаменационные билеты №1 (6.1)
	У4	Использует средства визуализации алгоритмов (блок-схемы)	Устный опрос (вопросы 9–10) (5.1); Тест №1 (5.3); Рефераты (темы 7–8) (5.2)	Экзаменационные билеты №2 (6.1)
	31	Знает понятие алгоритма, свойства и основные конструкции	Устный опрос (вопросы 1–6) (5.1); Тест №1 (5.3); Рефераты (темы 1–5) (5.2)	Экзаменационные билеты №1–2 (6.1)
	32	Классифицирует языки программирования	Устный опрос (вопросы 11–13) (5.1); Тест №1 (5.3); Рефераты (темы 11–15) (5.2)	Экзаменационные билеты №3 (6.1)
	33	Понимает назначение систем программирования	Устный опрос (вопросы 14–15) (5.1); Рефераты (темы 16–18) (5.2)	Экзаменационные билеты №3 (6.1)
Тема 2. Основные элементы языка. Управляющие операторы	У3	Реализует алгоритмы на языке программирования с использованием операторов и структур данных	Устный опрос (вопросы 16–25, 31–34) (5.1); Тест №2–3 (5.3);	Экзаменационные билеты №4–11 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
языка. Структурированные типы данных. Символьные типы данных			Рефераты (темы 22–35) (5.2)	
	У5	Оформляет программный код в соответствии со стандартами	Рефераты (тема 20) (5.2); Устный опрос (вопросы 15, 18) (5.1)	Экзаменационные билеты №24 (6.1)
	У6	Выполняет отладку программ и анализирует ошибки	Устный опрос (вопросы 30, 45) (5.1); Тест №2–3 (5.3); Рефераты (тема 19) (5.2)	Экзаменационные билеты №25 (6.1)
	34	Знает структуру программы и элементы языка	Устный опрос (вопросы 16–19) (5.1); Тест №2 (5.3); Рефераты (темы 21–24) (5.2)	Экзаменационные билеты №4 (6.1)
	35	Понимает методы реализации алгоритмов	Устный опрос (вопросы 33–35) (5.1); Тест №3 (5.3); Рефераты (темы 34–35) (5.2)	Экзаменационные билеты №10–11 (6.1)
	36	Понимает подпрограммы и их назначение	Устный опрос (вопросы 46–47) (5.1); Тест №4 (5.3); Рефераты (темы 48) (5.2)	Экзаменационные билеты №18–19 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
Тема 3. Модульное программирование. Рекурсия. Визуально-событийно управляемое программирование. Разработка оконного приложения	У3	Реализует программы с использованием функций, рекурсии и событийной модели	Устный опрос (вопросы 46–50) (5.1); Тест №4–5 (5.3); Рефераты (темы 48–50) (5.2)	Экзаменационные билеты №18–23 (6.1)
	У6	Выполняет тестирование и отладку программ	Устный опрос (вопрос 45) (5.1); Тест №4 (5.3); Рефераты (тема 19) (5.2)	Экзаменационные билеты №25 (6.1)
	36	Знает принципы модульного программирования	Устный опрос (вопрос 49) (5.1); Тест №4 (5.3); Рефераты (темы 48) (5.2)	Экзаменационные билеты №19 (6.1)
	37	Понимает принципы ООП и событийно-ориентированного программирования	Устный опрос (вопрос 50) (5.1); Тест №5 (5.3); Рефераты (темы 50) (5.2)	Экзаменационные билеты №21–23 (6.1)

3.2 Средства, применяемые для оценки уровня практической подготовки

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
Раздел 2. Архитектура и принципы работы основных логических блоков системы				
Практическое занятие № 1. Построение блок-схем	У1	Разрабатывает алгоритмы решения поставленных задач и представляет их в графической форме	Практическая работа № 1; Устный опрос (вопросы 1–	Экзаменационные билеты № 1–2 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
			10) (5.1); Тест № 1 (5.3)	
	У4	Использует средства проектирования для создания и графического отображения алгоритмов	Практическая работа № 1; Устный опрос (вопросы 9–10) (5.1); Рефераты (темы 7–8) (5.2)	Итоговая работа № 2 (6.1)
	31	Демонстрирует знание свойств алгоритмов, алгоритмических конструкций и способов их представления	Практическая работа № 1; Устный опрос (вопросы 1–10) (5.1); Тест № 1 (5.3)	Экзаменационные билеты № 1–2 (6.1)
Практическое занятие № 2. Операторы выбора, циклы	У1	Разрабатывает алгоритмы с использованием ветвлений и циклических конструкций	Практическая работа № 2; Устный опрос (вопросы 20–30) (5.1); Тест № 2 (5.3)	Экзаменационные билеты № 5–7 (6.1)
	У3	Реализует типовые алгоритмы с использованием условных операторов и циклов	Практическая работа № 2; Устный опрос (вопросы 20–30) (5.1); Тест № 2 (5.3)	Экзаменационные билеты № 5–7 (6.1)
	У6	Выполняет проверку и отладку программ с управляющими конструкциями	Практическая работа № 2; Устный опрос (вопросы 23, 29, 30) (5.1); Рефераты (тема 19) (5.2)	Итоговая работа № 25 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
	34	Демонстрирует знание основных элементов языка, операторов и структуры программы	Практическая работа № 2; Устный опрос (вопросы 18–30) (5.1); Тест № 2 (5.3)	Экзаменационные билеты № 4–7 (6.1)
Практическое занятие № 3. Работа с массивами	У2	Определяет эффективность способов обработки массивов и выбирает оптимальный алгоритм	Практическая работа № 3; Устный опрос (вопросы 31–35) (5.1); Рефераты (тема 10) (5.2)	Экзаменационные билеты № 8–11 (6.1)
	У3	Реализует программы обработки одномерных и двумерных массивов	Практическая работа № 3; Устный опрос (вопросы 31–35) (5.1); Тест № 3 (5.3)	Экзаменационные билеты № 8–11 (6.1)
	У6	Выполняет проверку, поиск и исправление ошибок в программах обработки массивов	Практическая работа № 3; Устный опрос (вопросы 33–35) (5.1); Тест № 3 (5.3)	Итоговая работа № 25 (6.1)
	35	Демонстрирует знание методов реализации типовых алгоритмов обработки и сортировки массивов	Практическая работа № 3; Устный опрос (вопросы 33–35) (5.1); Тест № 3 (5.3)	Экзаменационные билеты № 10–11 (6.1)
Практическое занятие № 4. Коллекции и	У3	Реализует программы с использованием коллекций и контейнеров	Практическая работа № 4; Устный опрос	Итоговая работа № 12 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
контейнеры			(вопросы 36–39) (5.1); Тест № 3 (5.3)	
	У5	Оформляет код программ с использованием структур данных в соответствии со стандартом кодирования	Практическая работа № 4; Рефераты (тема 20) (5.2)	Итоговая работа № 24 (6.1)
	35	Демонстрирует знание методов реализации операций над коллекциями и контейнерами	Практическая работа № 4; Устный опрос (вопросы 36–39) (5.1); Рефераты (темы 37–39) (5.2)	Итоговая работа № 12 (6.1)
Практическое занятие № 5. Символы и строки	У3	Реализует программы обработки символьных данных и строк	Практическая работа № 5; Устный опрос (вопросы 39–40) (5.1); Тест № 3 (5.3)	Экзаменационные билеты № 13–14 (6.1)
	У5	Оформляет код программ обработки строк в соответствии со стандартом кодирования	Практическая работа № 5; Рефераты (тема 20) (5.2)	Итоговая работа № 24 (6.1)
	34	Демонстрирует знание символьных типов данных и структуры программной обработки строк	Практическая работа № 5; Устный опрос (вопросы 39–40) (5.1); Рефераты (темы 41–44) (5.2)	Экзаменационные билеты № 13–14 (6.1)
Практическое занятие № 6. Работа с файлами	У3	Реализует программы чтения, записи и редактирования файлов	Практическая работа № 6; Устный опрос (вопросы	Экзаменационные билеты № 15–16 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
			41–45) (5.1); Тест № 3 (5.3)	
	У6	Выполняет проверку и отладку программ при работе с файлами и потоками данных	Практическая работа № 6; Устный опрос (вопросы 43–45) (5.1); Рефераты (тема 19) (5.2)	Итоговая работа № 25 (6.1)
	34	Демонстрирует знание средств языка для организации файлового ввода-вывода	Практическая работа № 6; Устный опрос (вопросы 41–45) (5.1); Рефераты (темы 45–47) (5.2)	Экзаменационные билеты № 15–16 (6.1)
Практическое занятие № 7. Реализация подпрограммы. Разработка рекурсивных подпрограмм	У3	Реализует модульную структуру программы с использованием процедур, функций и рекурсии	Практическая работа № 7; Устный опрос (вопросы 46–49) (5.1); Тест № 4 (5.3)	Экзаменационные билеты № 18–20 (6.1)
	У6	Выполняет проверку и отладку подпрограмм, выявляет ошибки при использовании рекурсии	Практическая работа № 7; Устный опрос (вопросы 48–49) (5.1); Тест № 4 (5.3)	Экзаменационные билеты № 20, 25 (6.1)
	36	Демонстрирует знание понятий подпрограммы, библиотеки подпрограмм, процедур, функций и рекурсии	Практическая работа № 7; Устный опрос (вопросы 46–49) (5.1); Рефераты (темы 48–	Экзаменационные билеты № 18–20 (6.1)

Краткое наименование раздела (модуля) / темы дисциплины	Код результата обучения	Показатель овладения результатами обучения	Наименование оценочного средства и представление его в КОС	
			Текущий контроль	Промежуточная аттестация
			49) (5.2)	
Практическое занятие № 8. Основные элементы управления. Разработка оконного приложения	У3	Реализует программный код оконного приложения с использованием элементов управления и обработки событий	Практическая работа № 8; Устный опрос (вопрос 50) (5.1); Тест № 5 (5.3)	Экзаменационные билеты № 21–23 (6.1)
	У5	Оформляет код оконного приложения в соответствии со стандартом кодирования	Практическая работа № 8; Рефераты (тема 20) (5.2)	Итоговая работа № 24 (6.1)
	37	Демонстрирует знание объектно-ориентированной модели и принципов событийно-ориентированного программирования	Практическая работа № 8; Тест № 5 (5.3); Рефераты (тема 50) (5.2)	Экзаменационные билеты № 21–23 (6.1)

4 Описание процедуры оценивания

Результаты обучения по профессиональному модулю, уровень сформированности компетенций оцениваются по четырём бальной шкале оценками: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Текущая аттестация по профессиональному модулю проводится с целью систематической проверки достижений обучающихся. Объектами оценивания являются: степень усвоения теоретических знаний, уровень овладения практическими умениями и навыками по всем видам учебной работы, качество выполнения самостоятельной работы, учебная дисциплина (активность на занятиях, своевременность выполнения различных видов заданий, посещаемость всех видов занятий по аттестуемой профессиональному модулю).

При проведении промежуточной аттестации оценивается достижение студентом запланированных по профессиональному модулю результатов обучения, обеспечивающих результаты освоения образовательной программы в целом.

Критерии оценивания устного ответа

5 баллов - ответ показывает прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа; умение приводить примеры современных проблем изучаемой области.

4 балла - ответ, обнаруживающий прочные знания основных процессов изучаемой предметной области, отличается глубиной и полнотой раскрытия темы; владение

терминологическим аппаратом; умение объяснять сущность, явлений, процессов, событий, делать выводы и обобщения, давать аргументированные ответы, приводить примеры; свободное владение монологической речью, логичность и последовательность ответа. Однако допускается одна - две неточности в ответе.

3 балла – ответ, свидетельствующий в основном о знании процессов изучаемой предметной области, отличающийся недостаточной глубиной и полнотой раскрытия темы; знанием основных вопросов теории; слабо сформированными навыками анализа явлений, процессов, недостаточным умением давать аргументированные ответы и приводить примеры; недостаточно свободным владением монологической речью, логичностью и последовательностью ответа. Допускается несколько ошибок в содержании ответа; неумение привести пример развития ситуации, провести связь с другими аспектами изучаемой области.

2 балла – ответ, обнаруживающий незнание процессов изучаемой предметной области, отличающийся неглубоким раскрытием темы; незнанием основных вопросов теории, несформированными навыками анализа явлений, процессов; неумением давать аргументированные ответы, слабым владением монологической речью, отсутствием логичности и последовательности. Допускаются серьезные ошибки в содержании ответа; незнание современной проблематики изучаемой области.

Критерии оценивания письменной работы

5 баллов - студент выразил своё мнение по сформулированной проблеме, аргументировал его, точно определив ее содержание и составляющие. Проблема раскрыта полностью, выводы обоснованы. Приведены данные отечественной и зарубежной литературы, статистические сведения, информация нормативно-правового характера. Студент владеет навыком самостоятельной работы по заданной теме; методами и приемами анализа теоретических и/или практических аспектов изучаемой области. Фактических ошибок, связанных с пониманием проблемы, нет; графически работа оформлена правильно.

4 балла - работа характеризуется смысловой цельностью, связностью и последовательностью изложения; допущено не более 1 ошибки при объяснении смысла или содержания проблемы. Проблема раскрыта. Не все выводы сделаны и/или обоснованы. Для аргументации приводятся данные отечественных и зарубежных авторов. Продемонстрированы исследовательские умения и навыки. Фактических ошибок, связанных с пониманием проблемы, нет. Допущены одна-две ошибки в оформлении работы.

3 балла – студент проводит достаточно самостоятельный анализ основных этапов и смысловых составляющих проблемы; понимает базовые основы и теоретическое обоснование выбранной темы. Проблема раскрыта не полностью. Выводы не сделаны и/или выводы не обоснованы. Проведен анализ проблемы без привлечения дополнительной литературы. Допущено не более 2 ошибок в смысле или содержании проблемы, оформлении работы.

2 балла - работа представляет собой пересказанный или полностью переписанный исходный текст без каких бы то ни было комментариев, анализа. Не раскрыта структура и теоретическая составляющая темы. Проблема не раскрыта. Выводы отсутствуют. Допущено три или более трех ошибок в смысловом содержании раскрываемой проблемы, в оформлении работы.

Критерии оценивания тестового задания

Оценка	<i>Отлично</i>	<i>Хорошо</i>	<i>Удовлетворительно</i>	<i>Неудовлетворительно</i>
Количество правильных ответов	90-100%	75-89%	60-74%	менее 60%

Критерии выставления оценки студенту на зачете/ экзамене

Оценка по промежуточной аттестации	Характеристика качества сформированности компетенций
«зачтено» / «отлично»	Студент демонстрирует сформированность дисциплинарных компетенций на продвинутом уровне: обнаруживает всестороннее, систематическое и глубокое знание учебного материала, умеет тесно увязывать теорию с практикой, свободно справляется с задачами, вопросами и другими видами применения знаний, причем не затрудняется с ответом при видоизменении заданий, правильно обосновывает принятое решение, владеет разносторонними навыками и приемами выполнения практических задач.
«зачтено» / «хорошо»	Студент демонстрирует сформированность дисциплинарных компетенций на базовом уровне: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.
«зачтено» / «удовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на пороговом уровне: имеет знания только основного материала, но не усвоил его деталей, в ходе контрольных мероприятий допускаются значительные ошибки, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, испытывает затруднения при выполнении практических работ, при оперировании знаниями и умениями при их переносе на новые ситуации.
«не зачтено» / «неудовлетворительно»	Студент демонстрирует сформированность дисциплинарных компетенций на уровне ниже порогового: выявляется полное или практически полное отсутствие знаний значительной части программного материала, студент допускает существенные ошибки, неуверенно, с большими затруднениями выполняет практические работы, умения и навыки не сформированы.

5. Примеры оценочных средств для проведения текущей аттестации

5.1 Вопросы для собеседования (устного опроса):

1. Что такое алгоритм?
2. Какие свойства алгоритма вы знаете?
3. Какие основные алгоритмические конструкции существуют?
4. Что такое линейный алгоритм?
5. Что такое разветвляющийся алгоритм?
6. Что такое циклический алгоритм?
7. Какие виды циклов существуют?
8. В чем разница между циклом с предусловием и постусловием?
9. Что такое блок-схема алгоритма?
10. Какие основные элементы используются в блок-схемах?
11. Что такое язык программирования?
12. Какие виды языков программирования существуют?
13. В чем разница между компилируемыми и интерпретируемыми языками?

14. Что такое система программирования?
15. Какие этапы включает процесс разработки программы?
16. Что такое переменная?
17. Какие типы данных вы знаете?
18. Что такое оператор в языке программирования?
19. Какие операторы ввода и вывода используются в языке программирования?
20. Что такое условный оператор?
21. Какие виды операторов выбора существуют?
22. Что такое оператор безусловного перехода?
23. В каких случаях используется оператор условного перехода?
24. Что такое оператор case (switch)?
25. В чем отличие оператора if от оператора case?
26. Что такое цикл?
27. В чем разница между циклами for, while и do-while?
28. Что такое вложенные циклы?
29. Что такое бесконечный цикл?
30. Как избежать заикливания программы?
31. Что такое массив?
32. Чем одномерный массив отличается от двумерного?
33. Какие операции можно выполнять над массивами?
34. Что такое сортировка массива?
35. Какие алгоритмы сортировки вы знаете?
36. Что такое коллекции и контейнеры?
37. В чем отличие массива от коллекции?
38. Какие операции выполняются над коллекциями?
39. Что такое строка?
40. Какие операции можно выполнять со строками?
41. Что такое файл в программировании?
42. Какие операции можно выполнять с файлами?
43. В чем разница между чтением и записью в файл?
44. Что такое поток данных?
45. Какие ошибки могут возникать при работе с файлами?
46. Что такое функция?
47. Чем функция отличается от процедуры?
48. Что такое рекурсия?
49. В чем преимущества модульного программирования?
50. Что такое событийно-ориентированное программирование?

5.2 Темы рефератов

1. Понятие алгоритма и его свойства
2. Основные алгоритмические конструкции
3. Линейные алгоритмы и их применение
4. Разветвляющиеся алгоритмы
5. Циклические алгоритмы
6. Виды циклов в алгоритмах
7. Блок-схемы алгоритмов и их элементы
8. Способы представления алгоритмов
9. Принципы построения алгоритмов
10. Оценка сложности алгоритмов
11. Понятие языка программирования
12. Классификация языков программирования
13. Эволюция языков программирования
14. Компилируемые языки программирования

15. Интерпретируемые языки программирования
16. Системы программирования
17. Среды разработки программ
18. Этапы разработки программного обеспечения
19. Отладка программ и методы поиска ошибок
20. Стандарты оформления программного кода
21. Структура программы
22. Основные элементы языка программирования
23. Переменные и типы данных
24. Операторы языка программирования
25. Операторы ввода и вывода данных
26. Условные операторы
27. Оператор выбора (if)
28. Оператор множественного выбора (case, switch)
29. Операторы перехода
30. Циклы в языках программирования
31. Массивы как структура данных
32. Одномерные массивы
33. Двумерные массивы
34. Основные операции над массивами
35. Алгоритмы сортировки массивов
36. Поиск элементов в массиве
37. Коллекции и контейнеры
38. Отличие массивов от коллекций
39. Операции над коллекциями
40. Многомерные структуры данных
41. Символьные типы данных
42. Строки и их обработка
43. Основные операции со строками
44. Работа со строковыми массивами
45. Файлы в программировании
46. Операции чтения и записи файлов
47. Потоки данных
48. Процедуры и функции
49. Рекурсия в программировании
50. Визуально-событийно управляемое программирование

5.3 Примеры тестовых заданий

Тест № 1

1. Какое из утверждений наиболее точно отражает свойство дискретности алгоритма?
 - а) алгоритм должен приводить к результату за конечное число шагов
 - б) алгоритм состоит из отдельных последовательно выполняемых команд (верно)
 - в) алгоритм должен быть применим к целому классу задач
 - г) алгоритм должен допускать только один способ записи
2. Какое требование относится именно к массовости алгоритма?
 - а) алгоритм должен быть понятен исполнителю
 - б) алгоритм должен быть записан в графической форме
 - в) алгоритм должен решать не одну конкретную задачу, а класс однотипных задач (верно)
 - г) алгоритм должен содержать не более трех базовых конструкций
3. Какой из перечисленных вариантов является наиболее корректным примером циклической структуры?

- а) выбор одного из нескольких действий в зависимости от условия
 - б) последовательное выполнение команд без проверок условий
 - в) многократное повторение группы команд до выполнения условия (верно)
 - г) передача управления на первую строку программы после ввода данных
4. Какой элемент блок-схемы обычно используется для обозначения проверки условия?
- а) прямоугольник
 - б) параллелограмм
 - в) ромб (верно)
 - г) овал
5. Какое утверждение о конечности алгоритма является правильным?
- а) алгоритм может не завершаться, если содержит цикл
 - б) конечность означает обязательное наличие команды завершения
 - в) конечность означает получение результата за конечное число шагов (верно)
 - г) конечность характерна только для численных алгоритмов
6. Что из перечисленного в наибольшей степени отличает интерпретируемый язык от компилируемого?
- а) программы на нем нельзя запускать повторно
 - б) выполнение программы обычно происходит без предварительного создания отдельного исполняемого файла машинного кода (верно)
 - в) он не поддерживает условные операторы
 - г) на нем невозможно работать с файлами
7. Какой вариант лучше всего описывает систему программирования?
- а) это только язык программирования и его синтаксис
 - б) это совокупность языка, транслятора, библиотек и средств разработки (верно)
 - в) это любая операционная система, в которой можно писать код
 - г) это только редактор исходного текста
8. В каком случае алгоритм нельзя считать определенным?
- а) если для разных исходных данных получаются разные результаты
 - б) если исполнитель может по-разному трактовать одну и ту же команду (верно)
 - в) если алгоритм записан в виде блок-схемы
 - г) если в алгоритме используется цикл
9. Что из перечисленного наиболее точно характеризует линейный алгоритм?
- а) в нем обязательно используется оператор выбора
 - б) команды выполняются последовательно, без ветвлений и повторений (верно)
 - в) он используется только для математических вычислений
 - г) он не может быть представлен блок-схемой
10. Какое утверждение о языках программирования является наиболее корректным?
- а) любой язык программирования относится только к низкому уровню
 - б) любой язык программирования обязательно компилируемый
 - в) классификация языков может выполняться по уровню, способу трансляции и области применения (верно)
 - г) различия между языками заключаются только в синтаксисе записи циклов

Время выполнения – 10 минут

1. Какое из утверждений наиболее точно описывает оператор присваивания?
 - а) он сравнивает два значения между собой
 - б) он передает управление другой части программы
 - в) он сохраняет вычисленное значение в переменной (верно)
 - г) он всегда изменяет тип данных переменной
2. В каком случае целесообразнее использовать оператор множественного выбора case/switch, а не цепочку if?
 - а) когда требуется проверить сложное логическое выражение с диапазонами
 - б) когда необходимо выбрать одно действие по одному из нескольких фиксированных значений выражения (верно)
 - в) когда нужно организовать бесконечный цикл
 - г) когда условие отсутствует
3. Какое утверждение наиболее точно описывает различие между while и do-while?
 - а) в do-while тело цикла может не выполниться ни разу
 - б) while используется только для счетных циклов
 - в) do-while гарантирует хотя бы одно выполнение тела цикла (верно)
 - г) while не допускает использования логического условия
4. Что является наиболее вероятным следствием неправильной формулировки условия завершения цикла?
 - а) программа обязательно завершится с ошибкой компиляции
 - б) программа может перейти в бесконечный цикл (верно)
 - в) программа всегда пропустит тело цикла
 - г) программа автоматически исправит условие
5. Какое утверждение об условном операторе является правильным?
 - а) он используется только с числовыми переменными
 - б) он позволяет изменить естественный порядок выполнения программы в зависимости от истинности условия (верно)
 - в) он может содержать только одну команду в теле
 - г) он не может быть вложенным
6. Что из перечисленного наиболее точно относится к логической ошибке программы?
 - а) пропущенная точка с запятой
 - б) использование необъявленной переменной
 - в) программа выполняется, но выдает неверный результат (верно)
 - г) отсутствие файла компилятора
7. Какой вариант лучше всего описывает назначение оператора безусловного перехода?
 - а) выполнение цикла с неизвестным числом повторений
 - б) немедленная передача управления указанной точке программы без проверки условия (верно)
 - в) проверка нескольких условий подряд
 - г) присваивание значения логической переменной
8. В каком случае вложенные условные операторы предпочтительнее одного простого условия?
 - а) когда решение зависит от нескольких взаимосвязанных проверок (верно)
 - б) когда программа содержит только одну переменную
 - в) когда необходимо выполнить многократный повтор действий
 - г) когда требуется только ввод данных

9. Какое утверждение о структуре программы является наиболее точным?
- а) программа может состоять только из операторов вывода
 - б) структура программы включает объявления, операторы и, в зависимости от языка, служебные конструкции (верно)
 - в) структура программы во всех языках абсолютно одинакова
 - г) наличие функции main обязательно в любом языке
10. В каком варианте наиболее корректно описано назначение операторов ввода-вывода?
- а) они служат только для сохранения данных в оперативной памяти
 - б) они обеспечивают взаимодействие программы с внешними источниками данных и пользователем (верно)
 - в) они нужны только при работе с графическим интерфейсом
 - г) они используются исключительно в интерпретируемых языках

Время выполнения – 10 минут

Тест № 3

1. Какое утверждение наиболее точно характеризует одномерный массив?
- а) это набор элементов разных типов, доступ к которым возможен только по имени
 - б) это упорядоченный набор однотипных элементов, доступ к которым осуществляется по индексу (верно)
 - в) это структура для хранения только символьных данных
 - г) это файл, содержащий несколько записей
2. Какое из действий не относится к базовой обработке массива?
- а) заполнение
 - б) сортировка
 - в) обход элементов
 - г) компиляция (верно)
3. В чем состоит ключевое отличие двумерного массива от одномерного?
- а) двумерный массив может хранить элементы только строкового типа
 - б) доступ к элементу двумерного массива осуществляется, как правило, по двум индексам (верно)
 - в) двумерный массив нельзя сортировать
 - г) двумерный массив всегда содержит больше данных, чем одномерный
4. Какой вариант наиболее точно описывает сортировку массива?
- а) это удаление повторяющихся элементов
 - б) это разбиение массива на несколько частей
 - в) это упорядочивание элементов по заданному правилу (верно)
 - г) это поиск минимального элемента без изменения порядка остальных
5. Какое утверждение о коллекциях по сравнению с массивами является наиболее корректным?
- а) коллекции всегда имеют фиксированный размер
 - б) коллекции вообще не поддерживают хранение объектов
 - в) коллекции часто предоставляют более гибкие операции добавления, удаления и обработки элементов (верно)
 - г) коллекции не допускают перебора элементов
6. Что обычно понимается под обработкой строки?
- а) только вывод строки на экран
 - б) только считывание строки с клавиатуры

- в) операции поиска, сравнения, разделения, объединения и преобразования символов (верно)
 - г) исключительно перевод строки в числовой тип
7. Какое из утверждений наиболее точно описывает файл в программировании?
- а) это область памяти, существующая только во время работы программы
 - б) это именованная область долговременного хранения данных (верно)
 - в) это команда языка для вывода текста
 - г) это разновидность массива
8. Что является важным условием корректной работы программы с файлом?
- а) файл всегда должен содержать только числа
 - б) перед выполнением операций чтения или записи файл должен быть открыт соответствующим способом (верно)
 - в) чтение из файла возможно только после записи
 - г) файл нельзя закрывать до завершения программы
9. Какое утверждение о потоках данных является наиболее правильным?
- а) поток всегда связан только с сетью Интернет
 - б) поток — это способ последовательной передачи данных между источником и приемником (верно)
 - в) поток используется только для работы с изображениями
 - г) поток полностью заменяет массивы и файлы
10. Какая ошибка наиболее вероятна при чтении данных из файла?
- а) обращение к несуществующему или недоступному файлу (верно)
 - б) автоматическое удаление файла компилятором
 - в) преобразование любого текста в исполняемый код
 - г) невозможность использовать переменные в программе

Время выполнения – 10 минут

Тест № 4

1. Какое утверждение наиболее точно отражает назначение подпрограмм?
- а) они используются только для графического интерфейса
 - б) они позволяют выделять законченные фрагменты решения задачи и повторно их использовать (верно)
 - в) они предназначены только для работы с массивами
 - г) они нужны исключительно для ускорения компиляции
2. В чем наиболее существенное различие между функцией и процедурой?
- а) функция всегда работает быстрее процедуры
 - б) функция обычно возвращает значение, а процедура может не возвращать его явно (верно)
 - в) процедура не может иметь параметры
 - г) функция не может содержать условные операторы
3. Что обычно понимается под параметрами подпрограммы?
- а) команды, которые выполняются после завершения программы
 - б) данные, передаваемые в подпрограмму для обработки (верно)
 - в) глобальные переменные программы
 - г) служебные слова языка, определяющие тип файла
4. Какое утверждение о локальных переменных является наиболее точным?

- а) они доступны из любой части программы
- б) они существуют только в пределах той подпрограммы или блока, где объявлены (верно)
- в) они обязательно сохраняют значение после завершения программы
- г) они заменяют все параметры функций

5. Что из перечисленного является преимуществом модульного программирования?

- а) отказ от использования переменных
- б) устранение необходимости тестирования
- в) повышение читаемости, повторного использования и удобства сопровождения кода (верно)
- г) возможность писать программу без алгоритма

6. Какое утверждение наиболее точно описывает рекурсию?

- а) это способ записи цикла только через оператор for
- б) это вызов подпрограммы самой себя прямо или косвенно (верно)
- в) это обязательный элемент объектно-ориентированного программирования
- г) это ошибка, возникающая при неправильной работе с памятью

7. При каком условии рекурсивная подпрограмма может корректно завершаться?

- а) если она не имеет параметров
- б) если в ней отсутствуют вычисления
- в) если задано условие выхода из рекурсии (верно)
- г) если она вызывается только один раз

8. Что наиболее вероятно произойдет при отсутствии базового случая в рекурсивной функции?

- а) функция автоматически станет итерационной
- б) программа может завершиться переполнением стека вызовов (верно)
- в) рекурсия будет проигнорирована компилятором
- г) функция обязательно вернет ноль

9. Какой вариант наиболее точно описывает библиотеку подпрограмм?

- а) набор случайных операторов из разных программ
- б) совокупность готовых процедур и функций, предназначенных для многократного использования (верно)
- в) файл, содержащий только комментарии к коду
- г) среда выполнения программы

10. Какое утверждение о проверке и отладке программ наиболее корректно?

- а) отладка нужна только для синтаксических ошибок
- б) проверка программы включает поиск синтаксических, логических и runtime-ошибок (верно)
- в) после компиляции отладка не требуется
- г) тестирование и отладка означают одно и то же

Время выполнения – 10 минут

Тест № 5

1. Что наиболее точно характеризует событийно-ориентированное программирование?

- а) программа выполняется только последовательно от первой строки к последней без внешнего воздействия
- б) выполнение отдельных частей программы инициируется наступлением событий (верно)
- в) программа может работать только в браузере

г) такой подход применяется только к системному программированию

2. Что в графическом приложении обычно понимается под событием?

- а) любой элемент исходного кода
- б) действие пользователя или системы, на которое программа может реагировать (верно)
- в) только ошибка времени выполнения
- г) только запуск программы

3. Какое из перечисленных понятий наиболее тесно связано с виджетами?

- а) структуры сортировки
- б) элементы пользовательского интерфейса (верно)
- в) типы компиляторов
- г) машинные команды процессора

4. Что является примером элемента управления в оконном приложении?

- а) цикл while
- б) массив
- в) кнопка (верно)
- г) оператор присваивания

5. Какое утверждение наиболее точно описывает класс в объектно-ориентированном программировании?

- а) это конкретный экземпляр объекта в памяти
- б) это шаблон, описывающий свойства и методы объектов (верно)
- в) это оператор выбора одной из ветвей программы
- г) это специальный вид файла для хранения данных

6. Что наиболее точно понимается под объектом?

- а) алгоритм сортировки массива
- б) экземпляр класса, обладающий состоянием и поведением (верно)
- в) только графический элемент интерфейса
- г) тип ошибки компиляции

7. Какое утверждение об инкапсуляции является наиболее корректным?

- а) это обязательное использование рекурсии внутри класса
- б) это объединение данных и методов работы с ними в одной сущности с ограничением прямого доступа к внутреннему состоянию (верно)
- в) это возможность существования нескольких классов с одинаковым именем
- г) это механизм многократного запуска цикла

8. Какое из утверждений лучше всего описывает наследование?

- а) это способ передачи параметров в функцию
- б) это механизм создания нового класса на основе существующего с заимствованием его свойств и методов (верно)
- в) это способ хранения данных в файле
- г) это обязательное наличие нескольких конструкторов

9. В чем заключается полиморфизм?

- а) в возможности использовать один интерфейс для объектов разных классов с различной реализацией поведения (верно)
- б) в запрете изменения методов базового класса
- в) в объединении всех объектов в один класс
- г) в хранении разных типов данных в массиве

10. Что наиболее важно при разработке простого оконного приложения?

- а) наличие только одного файла исходного кода
- б) организация взаимодействия элементов интерфейса и обработчиков событий (верно)
- в) обязательное использование рекурсии
- г) отсутствие пользовательского ввода

Время выполнения – 10 минут

6. Примеры оценочных средств для проведения промежуточной аттестации

6.1 Варианты экзаменационных билетов

Итоговая работа №1

- 1. Понятие алгоритма и его свойства
- 2. Линейные и разветвляющиеся алгоритмы

Практическое задание:

Разработать алгоритм и реализовать программу, которая принимает число N и вычисляет сумму чисел от 1 до N. Вывести результат.

Итоговая работа №2

- 1. Циклические алгоритмы и виды циклов
- 2. Блок-схемы алгоритмов

Практическое задание:

Построить блок-схему и написать программу, которая выводит все четные числа от 1 до 100.

Итоговая работа №3

- 1. Классификация языков программирования
- 2. Компилируемые и интерпретируемые языки

Практическое задание:

Написать программу, которая принимает число и определяет, является ли оно простым.

Итоговая работа №4

- 1. Структура программы
- 2. Переменные и типы данных

Практическое задание:

Написать программу, которая запрашивает у пользователя два числа и выводит их сумму, разность, произведение и частное.

Итоговая работа №5

1. Операторы ввода и вывода
2. Оператор if и его применение

Практическое задание:

Написать программу, которая принимает число и определяет, положительное оно, отрицательное или равно нулю.

Итоговая работа №6

1. Операторы выбора
2. Оператор case (switch)

Практическое задание:

Написать программу, которая принимает номер дня недели (1–7) и выводит его название.

Итоговая работа №7

1. Циклы for, while, do-while
2. Вложенные циклы

Практическое задание:

Написать программу, которая выводит таблицу умножения от 1 до 10.

Итоговая работа №8

1. Понятие массива
2. Одномерные массивы

Практическое задание:

Создать массив из 10 элементов, заполнить его числами и найти максимальный элемент.

Итоговая работа №9

1. Двумерные массивы
2. Основные операции над массивами

Практическое задание:

Создать двумерный массив 3×3 , заполнить числами и найти сумму всех элементов.

Итоговая работа №10

1. Сортировка массивов
2. Алгоритмы сортировки

Практическое задание:

Реализовать сортировку массива из 10 элементов по возрастанию.

Итоговая работа №11

1. Поиск в массиве
2. Линейный и бинарный поиск

Практическое задание:

Написать программу поиска заданного числа в массиве.

Итоговая работа №12

1. Коллекции и контейнеры
2. Отличие коллекций от массивов

Практическое задание:

Создать коллекцию чисел, добавить элементы и вывести их на экран.

Итоговая работа №13

1. Символьные типы данных
2. Строки и их обработка

Практическое задание:

Написать программу, которая принимает строку и выводит количество символов.

Итоговая работа №14

1. Основные операции со строками
2. Строковые массивы

Практическое задание:

Написать программу, которая проверяет, является ли строка палиндромом.

Итоговая работа №15

1. Файлы в программировании
2. Чтение и запись файлов

Практическое задание:

Создать файл, записать в него строку и затем считать её и вывести на экран.

Итоговая работа №16

1. Потоки данных
2. Работа с файлами

Практическое задание:

Написать программу, которая считывает числа из файла и выводит их сумму.

Итоговая работа №17

1. Локальные и глобальные переменные
2. Область видимости переменных

Практическое задание:

Создать программу с глобальной и локальной переменной и продемонстрировать их различие.

Итоговая работа №18

1. Процедуры и функции
2. Передача параметров

Практическое задание:

Создать функцию, которая принимает два числа и возвращает их сумму.

Итоговая работа №19

1. Модульное программирование
2. Библиотеки подпрограмм

Практическое задание:

Разделить программу на функции: ввод данных, обработка и вывод результата.

Итоговая работа №20

1. Рекурсия
2. Примеры рекурсивных алгоритмов

Практическое задание:

Написать рекурсивную функцию вычисления факториала числа.

Итоговая работа №21

1. Событийно-ориентированное программирование
2. Обработка событий

Практическое задание:

Создать программу с кнопкой, при нажатии на которую выводится сообщение.

Итоговая работа №22

1. Виджеты и элементы управления
2. Интерфейс пользователя

Практическое задание:

Создать окно с полем ввода и кнопкой, при нажатии выводить введенный текст.

Итоговая работа №23

1. Разработка оконного приложения
2. Основные элементы управления

Практическое задание:

Создать приложение с двумя полями ввода и кнопкой для вычисления суммы.

Итоговая работа №24

1. Установка и запуск программ
2. Этапы разработки приложения

Практическое задание:

Создать программу и описать шаги её компиляции и запуска.

Итоговая работа №25

1. Отладка программ
2. Виды ошибок в программировании

Практическое задание:

Найти и исправить ошибки в программе, которая неправильно вычисляет сумму массива.