

**МИНОБРНАУКИ РОССИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**  
**ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ»**

**УТВЕРЖДАЮ**  
Проректор по учебной работе

\_\_\_\_\_ Василенко В.Н.

«25» мая 2023

**РАБОЧАЯ ПРОГРАММА**  
**ДИСЦИПЛИНЫ**

**Алгоритмы и структуры данных**  
(наименование в соответствии с РУП)

Направление подготовки (специальность)

10.05.03 Информационная безопасность автоматизированных систем  
(шифр и наименование направления подготовки/специальности)

Специализация

Безопасность открытых информационных систем  
(наименование профиля/специализации)

Квалификация выпускника  
Квалификация: специалист по защите информации

Воронеж

### 1. Цели и задачи дисциплины

Целью освоения дисциплины «Алгоритмы и структуры данных» является формирование компетенций обучающегося в области профессиональной деятельности и сфере профессиональной деятельности:

- 06 Связь, информационные и коммуникационные технологии (в сфере обеспечения безопасности информации в автоматизированных системах).

Дисциплина направлена на решение задач профессиональной деятельности научно-исследовательского, проектного, контрольно-аналитического, эксплуатационного типов.

Программа составлена в соответствии с требованиями Федерального государственного образовательного стандарта высшего образования по специальности

10.05.03 Информационная безопасность автоматизированных систем.

### 2. Перечень планируемых результатов обучения, соотнесенных с планируемыми результатами освоения образовательной программы

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	ОПК-7	Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ИД1опк-7 – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 <sub>опк-7</sub> – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности	Знает: - основные виды структур данных; алгоритмы обработки информации, хранящейся в различных видах структур данных; достоинства и недостатки каждого вида структур данных для применения при решении различных задач.
	Умеет: - делать обоснованный выбор используемых при решении задач структур данных; применять структуры данных и алгоритмы их обработки при решении различных задач; применять полученные знания и навыки в своей дальнейшей профессиональной деятельности.
	Владеет: навыками создания собственных и использования библиотечных структур данных при разработке программ на языке программирования C++.

### 3. Место дисциплины (модуля) в структуре ООП ВО/СПО

Дисциплина относится к обязательной части Блока 1 ООП. Дисциплина является обязательной к изучению.

Изучение дисциплины основано на знаниях, умениях и навыках, полученных при изучении обучающимися дисциплины информатика на базе среднего или средне-специального образования.

Дисциплина является предшествующей для изучения последующих дисциплин, Технологии и методы программирования, Параллельное программирование.

### 4. Объем дисциплины (модуля) и виды учебной работы

Общая трудоемкость дисциплины (модуля) составляет 3 зачетных единиц.

Виды учебной работы	Всего академических часов	Распределение трудоемкости по семестрам, ак. ч
		1 семестр
Общая трудоемкость дисциплины (модуля)	<b>108</b>	<b>108</b>
<b>Контактная работа</b> в т. ч. аудиторные занятия:	<b>61,6</b>	<b>61,6</b>
Лекции	30	30
в том числе в форме практической подготовки	–	–
Практические занятия	30	30
в том числе в форме практической подготовки	–	–
Текущие консультации	1,5	1,5
<b>Вид аттестации – зачет</b>	0,1	0,1
<b>Самостоятельная работа:</b>	<b>46,4</b>	<b>46,4</b>
Проработка материалов по лекциям, учебникам, учебным пособиям	15	15
Подготовка к практическим занятиям	15	15
Домашнее задание	16,4	16,4

**5 Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий**

### 5.1 Содержание разделов дисциплины (модуля)

№ п/п	Наименование раздела дисциплины	Содержание раздела (указываются темы и дидактические единицы)	Трудоемкость раздела, ак.ч
1	Введение. Структурированные типы данных. Подпрограммы.	Язык С++: алфавит, ключевые слова, лексемы, стандартные типы данных. Структура программы на С++. Стандартные типы данных. Библиотеки С++. Описание типа «массив». Одномерные и многомерные массивы. Правила работы с массивами. Создание и обработка массивов. Строки. Файлы. Записи. Понятие подпрограммы. Подпрограмма-функция, описание функций. Область действия идентификаторов.	50
2	Базовые алгоритмы решения задач.	Алгоритмы поиска. Алгоритмы сортировки: внутренняя и внешняя сортировка. Использование хеширования для поиска данных. Решение задач с перебором: алгоритмы типа "разделяй и властвуй", метод динамического программирования.	26
3	Динамические структуры данных.	Определение структуры данных. Линейные связанные списки: однонаправленные и двунаправленные. Очередь, стек, реализации в виде массива и списка. Примеры приложений, использующих списки, стеки и очереди. Граф как структура данных. Бинарные деревья	30,4

### 5.2 Разделы дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Лекции, ак. ч	Практические, ак. ч	СРО, ак. ч
1	Введение. Структурированные типы данных. Подпрограммы.	16	14	20
2	Базовые алгоритмы решения задач.	6	8	12
3	Динамические структуры данных.	8	8	14,4

#### 5.2.1 Лекции

№ п/п	Наименование раздела дисциплины	Тематика лекционных занятия	Трудоемкость, ак. ч
1	Введение. Структурированные типы данных. Подпрограммы.	Язык С++: алфавит, ключевые слова, лексемы, стандартные типы данных. Структура программы на С++. Стандартные типы данных. Библиотеки С++. Описание типа «массив». Одномерные и многомерные массивы. Правила работы с массивами. Строки. Файлы. Записи. Понятие подпрограммы. Подпрограмма-функция, описание функций. Рекурсии	16
2	Базовые алгоритмы решения задач.	Алгоритмы поиска. Алгоритмы сортировки. Использование хеширования для поиска данных.	6
3	Динамические структуры данных.	Определение структуры данных. Линейные связанные списки: однонаправленные и двунаправленные. Очередь, стек, реализации в виде массива и списка. Граф как структура данных. Бинарные деревья	8

### 5.2.2 Практические занятия (семинары)

№ п/п	Наименование раздела дисциплины	Тематика практических занятия	Трудоемкость, ак. ч
1	Введение. Структурированные типы данных. Подпрограммы.	Структура программы. Компиляция и отладка программы на С++. Решение задач по теме линейные алгоритмы и алгоритмы ветвления. Решение задач по теме одномерные массивы. Решение задач по теме двумерные массивы. Решение задач по теме строки. Решение задач по теме динамические массивы. Решение задач по теме файлы. Решение задач по теме записи. Решение задач по теме подпрограммы, рекурсии.	14
2	Базовые алгоритмы решения задач.	Алгоритмы поиска. Алгоритмы сортировки: внутренняя и внешняя сортировка. Использование хеширования для поиска данных. Решение задач с перебором: алгоритмы типа "разделяй и властвуй", метод динамического программирования.	8
3	Динамические структуры данных.	Решение задач на тему "Динамические структуры данных - списки, стеки, очереди". Решение задач на тему бинарные деревья.	8

### 5.2.3 Лабораторный практикум не предусмотрен

### 5.2.4 Самостоятельная работа обучающихся

№ п/п	Наименование раздела дисциплины	Вид СРО	Трудоемкость, ак. ч
1	Введение. Структурированные типы данных. Подпрограммы. Базовые алгоритмы решения задач.	Проработка материалов по лекциям, учебникам, учебным пособиям, подготовка к тестированию	6
		Подготовка к практическим занятиям	6
		Домашнее задание	8
2	Динамические структуры данных.	Проработка материалов по лекциям, учебникам, учебным пособиям, подготовка к тестированию	4

		Подготовка к практическим занятиям	4
		Домашнее задание	4
3	Динамические структуры данных.	Проработка материалов по лекциям, учебникам, учебным пособиям, подготовка к тестированию	4
		Подготовка к практическим занятиям	4
		Домашнее задание	6,4

## 6 Учебно-методическое и информационное обеспечение дисциплины (модуля)

Для освоения дисциплины обучающийся может использовать:

### 6.1 Основная литература

1. Пантелеев, Е. Р. Алгоритмы и структуры данных : учебное пособие / Е. Р. Пантелеев, А. Л. Алыкова. — Иваново : ИГЭУ, 2018. — 142 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/154576>

2. Павлов, Л. А. Структуры и алгоритмы обработки данных : учебник для вузов / Л. А. Павлов, Н. В. Первова. — 3-е изд., стер. — Санкт-Петербург : Лань, 2021. — 256 с. — ISBN 978-5-8114-7259-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/156929>

3. Маер, А. В. Введение в структуры и алгоритмы обработки данных : учебное пособие / А. В. Маер, О. С. Черепанов. — Курган : КГУ, 2021. — 107 с. — ISBN 978-5-4217-0576-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/177907>

### 6.2 Дополнительная литература

1. Гулаков, В. К. Структуры и алгоритмы обработки многомерных данных : монография / В. К. Гулаков, А. О. Трубаков, Е. О. Трубаков. — Санкт-Петербург : Лань, 2021. — 356 с. — ISBN 978-5-8114-2962-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/169211>

2. Сыромятников, В. П. Структуры и алгоритмы обработки данных: Практикум : учебное пособие / В. П. Сыромятников. — Москва : РТУ МИРЭА, 2020. — 244 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/163915>

3. Апанасевич, С. А. Структуры и алгоритмы обработки данных. Линейные структуры : учебное пособие / С. А. Апанасевич. — Санкт-Петербург : Лань, 2019. — 136 с. — ISBN 978-5-8114-3366-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/113934>

### 6.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся

Алгоритмы и структуры данных: методические указания для самостоятельной работы для студентов, обучающихся по направлению 10.05.03 Информационная безопасность автоматизированных систем, очной формы обучения / А.В. Скрыпников, И. А. Высоцкая; ВГУИТ, Кафедра информационной безопасности. - Воронеж : ВГУИТ, 2021. - 13 с.

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

1. Сайт научной библиотеки ВГУИТ <<http://cnit.vsuet.ru>>.
2. Электронно-библиотечная система ЛАНЬ <<https://e.lanbook.com>>.
3. ЭБС «Университетская библиотека Онлайн» <<https://biblioclub.ru/>>.
4. Базовые федеральные образовательные порталы. <[http://www.edu.ru/db/portal/sites/portal\\_page.htm](http://www.edu.ru/db/portal/sites/portal_page.htm)>.
5. Государственная публичная научно-техническая библиотека. <[www.gpntb.ru/](http://www.gpntb.ru/)>.

6. Информационно-коммуникационные технологии в образовании. Система федеральных образовательных порталов. <<http://www.ict.edu.ru/>>.
7. Национальная электронная библиотека. <[www.nns.ru/](http://www.nns.ru/)>..
8. Поисковая система «Апорт». <[www.aport.ru/](http://www.aport.ru/)>.
9. Поисковая система «Рамблер». <[www.rambler.ru/](http://www.rambler.ru/)>.
10. Поисковая система «Yahoo» . <[www.yahoo.com/](http://www.yahoo.com/)>.
11. Поисковая система «Яндекс». <[www.yandex.ru/](http://www.yandex.ru/)>.
12. Российская государственная библиотека. <[www.rsl.ru/](http://www.rsl.ru/)>.
13. Российская национальная библиотека. <[www.nlr.ru/](http://www.nlr.ru/)>.

**6.5 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем**

Лекционные аудитории, оснащенные мультимедийной техникой	Ауд. 401: Аудио-визуальная система лекционных аудиторий (мультимедийный проектор, экран, усилитель мощности звука, акустические системы, микрофоны, устройство коммутации, сетевой коммутатор для подключения к компьютерной сети (Интернет))	
Аудитории для проведения практических занятий	Ауд. №332а: комп. класс каф. ИнфБ, количество ПЭВМ-12 (компьютер Cjrei5-4570, ауд.№ 420: комп. класс каф. ИнфБ, количество ПЭВМ - 12,(рабочая станция CPUCore 2DuoE6300 – 1.86), ауд. №424, комп класс каф. ИнфБ, количество ПЭВМ -12 (Компьютер Celeron D 2.8)	ОС Alt Linux (Альт Образование 8.2) Geany. Lazarus. Qt Creator. Quanta Plus. Веб-редактор Bluefish. Среда разработки Code::Blocks. Офисный пакет Libre Office 5.4: Base, Calc, Draw, Impress, Math, Writer. Персональная бухгалтерия HomeBank. Словарь Star Dict. iTest. VM Maxima. Кумир. Avidemux. Audacios. Brasero. Cheese. SMPlayer. Медиаплеер Parole. Редактор тегов Easy TAG. Stath Studio. Pinta. Веб-браузер Mozilla Firefox. Графический редактор. FP – free Pascal. Microsoft Windows 7 (64 разрядная) Microsoft Office (standart) 2007; Microsoft Access 2007; Microsoft Project 2007; Microsoft Share Point 2007; Microsoft Visio 2007; Mi-

		<p>Microsoft SQL server 2008; 1 С Предприятие Лицензия; 7-Zip File Manager (архиватор); Adobe Acrobat Reader; Adobe Flash Player; FAR file manager; Google Chrome; Java TM 7 (64-bit); K-Lite Codec Pack; Mozilla Firefox; Oracle VM VirtualBox; Sublime Text; Symantec Endpoint Protection 12 (Заменен на AVP Kaspersky); VMWare Player; Антивирус "Зоркий глаз"; Lazarus; SmathStudio; NanoCAD; Gimp (графический редактор, аналог Photoshop); Avidemux (видео редактор); Virtual Dub (видео редактор); Free Pascal; Страж NT вер.3.0 Сертификат ФСТЭК № 2145 30.07.2013 г.; Ревизор 1XP Сертификат ФСТЭК № 989 08.02.2015 г.; Ревизор 2XP Сертификат ФСТЭК № 990 08.02.2015 г.; Фикс 2.0.2 Сертификат ФСТЭК №1548 15.01.2015 г.; Ревизор сети вер.3.0 Сертификат ФСТЭК №3413 02.06.2015 г.; СЗИ Панцирь К Сертификат ФСТЭК №1973 09.12.2015 г.; СЗИ Dallas Lock 8.0 К Сертификат ФСТЭК №2720 25.09.2015; СЗИ Dallas Lock 8.0 С Сертификат ФСТЭК №2945 16.08.2013</p>
Аудитория для самостоятельной работы студентов (Читальные залы библиотеки)	Компьютеры со свободным доступом в сеть Интернет и Электронным библиотечным и информационно-справочным системам	
Аудитория для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Комплекты мебели для учебного процесса – 30 шт., доска	

**При освоении дисциплины используется лицензионное и открытое программное обеспечение – ОС Microsoft Windows, ОС ALT Linux, Microsoft Office Professional Plus; VMWare Player, Oracle VM VirtualBox, Visual Studio 2022.**

## Материально-техническое обеспечение дисциплины

<p>Учебная лаборатория для проведения практических занятий</p>	<p><b>Ауд. 424</b> Компьютер РЕГАРД – 11 шт., стенды – 3 шт.</p>	<p>ОС Astra Linux Альт Образование 8.2 [Лицензия № AAA.0217.00 с 21.12.2017 г. Лицензионный договор № РБТ-14/1623-01-ВУЗ от 18.12.2017 г.] бессрочно, Libre Office 6.1 [Лицензия № AAA.0217.00 с 21.12.2017 г. Включен в установочный пакет операционной системы Альт Образование 8.2] бессрочно, wxMaxima [Лицензия № AAA.0217.00 с 21.12.2017 г. ] бессрочно, Lazarus [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/Lazarus">https://ru.wikipedia.org/wiki/Lazarus</a>] бессрочно, Oracle VM Virtual Box [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/VirtualBox">https://ru.wikipedia.org/wiki/VirtualBox</a>] бессрочно, FreePascal [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/Free_Pascal">https://ru.wikipedia.org/wiki/Free_Pascal</a>] бессрочно.</p>
<p>Учебная лаборатория для проведения практических занятий</p>	<p><b>Ауд. 332а</b> Компьютеры - 12 шт., стенды – 5 шт.</p>	<p>ОС Astra Linux Альт Образование 8.2 [Лицензия № AAA.0217.00 с 21.12.2017 г. Лицензионный договор № РБТ-14/1623-01-ВУЗ от 18.12.2017 г.] бессрочно, Libre Office 6.1 [Лицензия № AAA.0217.00 с 21.12.2017 г. Включен в установочный пакет операционной системы Альт Образования 8.2] бессрочно , wxMaxima [Лицензия № AAA.0217.00 с 21.12.2017 г. Включен в установочный пакет операционной системы Альт Образование 8.2] бессрочно, Lazarus [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/Lazarus">https://ru.wikipedia.org/wiki/Lazarus</a>] бессрочно, SMathStudio [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/SMath_Studio">https://ru.wikipedia.org/wiki/SMath_Studio</a>] бессрочно, Avidemux [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/Avidemux">https://ru.wikipedia.org/wiki/Avidemux</a>] бессрочно, Oracle VM Virtual Box [ <a href="https://ru.wikipedia.org/wiki/VirtualBox">https://ru.wikipedia.org/wiki/VirtualBox</a>] бессрочно, AnyLogic 8.3 [(бесплатное ПО) <a href="https://www.anylogic.ru/downloads/personal-learning-edition-download/">https://www.anylogic.ru/downloads/personal-learning-edition-download/</a>] бессрочно.</p>
	<p>Ауд. 332 Компьютеры - 12 шт.</p>	<p>Альт Образование 8.2 [Лицензия № AAA.0217.00 с 21.12.2017 г. по «Бессрочно»], Libre Office 6.1 [Лицензия № AAA.0217.00 с 21.12.2017 г. по «Бессрочно» (Включен в установочный пакет операционной системы Альт Образование 8.2), wxMaxima [Лицензия № AAA.0217.00 с 21.12.2017 г. по «Бессрочно» (Включен в установочный пакет операционной системы Альт Образование 8.2), SMathStudio [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/SMath_Studio">https://ru.wikipedia.org/wiki/SMath_Studio</a>], Lazarus [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/Lazarus">https://ru.wikipedia.org/wiki/Lazarus</a>], Avidemux [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/Avidemux">https://ru.wikipedia.org/wiki/Avidemux</a>], Oracle VM Virtual Box [(бесплатное ПО) <a href="https://ru.wikipedia.org/wiki/VirtualBox">https://ru.wikipedia.org/wiki/VirtualBox</a>], AnyLogic 8.3 [(бесплатное ПО) </p>

		learning-edition-download/]
Аудитории для проведения занятий лекционного типа,	<b>Ауд. 401</b> Аудио-визуальная система лекционных аудитория (мультимедийный проектор Epson EB-X18, настенный экран Screen Media)	
Аудитории для самостоятельной работы, курсового и дипломного проектирования	<b>Читальные залы</b> библиотеки: Компьютеры со свободным доступом в сеть Интернет и Электронными библиотечными и информационно справочными системами, Компьютеры Regard - 12 шт.  <b>Ауд. 424</b> Компьютер РЕГАРД – 11 шт., стенды – 3 шт	АЛьт 8.1

### **8 Оценочные материалы для промежуточной аттестации обучающихся по дисциплине (модулю)**

**Оценочные материалы (ОМ)** для дисциплины (модуля) включают в себя:

- перечень компетенций с указанием индикаторов достижения компетенций, этапов их формирования в процессе освоения образовательной программы;
- описание шкал оценивания;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков;
- методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности.

ОМ представляются отдельным комплектом и **входят в состав рабочей программы дисциплины (модуля)**.

Оценочные материалы формируются в соответствии с П ВГУИТ «Положение об оценочных материалах».

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ  
ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

по дисциплине

**АЛГОРИТМЫ И СТРУКТУРЫ ДАННЫХ**

## 1 Перечень компетенций с указанием этапов их формирования

№ п/п	Код компетенции	Формулировка компетенции	Код и наименование индикатора достижения компетенции
1	ОПК-7	Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ	ИД1 <sub>опк-7</sub> – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности

Код и наименование индикатора достижения компетенции	Результаты обучения (показатели оценивания)
ИД1 <sub>опк-7</sub> – обладает способностью применять языки, системы и инструментальные средства программирования в профессиональной деятельности	Знает: основные виды структур данных; алгоритмы обработки информации, хранящейся в различных видах структур данных; достоинства и недостатки каждого вида структур данных для применения при решении различных задач.
	Умеет: делать обоснованный выбор используемых при решении задач структур данных; применять структуры данных и алгоритмы их обработки при решении различных задач; применять полученные знания и навыки в своей дальнейшей профессиональной деятельности.
	Владеет: навыками создания собственных и использования библиотечных структур данных при разработке программ на языке программирования C++.

## 2 Паспорт оценочных материалов по дисциплине

№ п/п	Разделы дисциплины	Индекс контролируемой компетенции	Оценочные средства		Технология/процедура оценивания (способ контроля)
			наименование	№№ заданий	
1.	Введение. Структурированные типы данных. Подпрограммы	ОПК-7	Банк тестовых заданий	1-31	Бланочное тестирование (процентная шкала)
			Домашнее задание	51-106	Проверка преподавателем (уровневая шкала)
			Собеседование (зачет)	131-139	Проверка преподавателем (уровневая шкала)
2.	Базовые алгоритмы решения задач.	ОПК-7	Банк тестовых заданий	107-115	Бланочное тестирование (процентная шкала)
			Домашнее задание	31-40	Проверка преподавателем (уровневая шкала)

			Собеседование зачет	140-153	Проверка преподавателем (уровневая шкала)
3.	Динамические структуры данных.	ОПК-7	Банк тестовых заданий	44-50	Бланочное тестирование (процентная шкала)
			Домашнее задание	116-130	Проверка преподавателем (уровневая шкала)
			Собеседование	153-155	Проверка преподавателем (уровневая шкала)

### 3 Оценочные материалы для промежуточной аттестации

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

Аттестация обучающегося по дисциплине проводится в форме тестирования и предусматривает возможность последующего собеседования (зачета).

Каждый вариант теста включает 15 контрольных заданий.

- 5 тестовых вопросов на проверку знаний;
- 10 тестовых вопросов на проверку умений и навыков;

Каждый билет включает 2 контрольных вопроса, из них:

- 1 контрольный вопрос на проверку знаний;
- 1 контрольный вопрос на проверку умений и навыков.

#### 3.1 Тесты (тестовые задания)

**3.1.1 Шифр и наименование компетенции ОПК-7** Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

<b>№ задания</b>	<b>Тестовое задание с вариантами ответов и правильными ответами</b>
1.	Алгоритм — это: <ul style="list-style-type: none"> <li>– указание на выполнение действий приводящее к заданному результату</li> <li>– процесс выполнения вычислений, приводящих к решению задачи.</li> <li>– система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи</li> <li>– последовательность действий</li> </ul>

2.	<p>Алгоритм может быть задан следующими способами: (несколько ответов)</p> <ul style="list-style-type: none"> <li>– <b>словесным</b></li> <li>– <b>на алгоритмическом языке</b></li> <li>– <b>графическим</b></li> <li>– <b>формально-словесным</b></li> <li>– словесно-графическим</li> <li>– последовательностью байтов.</li> </ul>
3.	<p>Программа — это:</p> <ul style="list-style-type: none"> <li>– система правил, описывающая последовательность действий, которые необходимо выполнить для решения задачи</li> <li>– указание на выполнение действий из заданного набора</li> <li>– область внешней памяти для хранения текстовых, числовых данных и другой информации</li> <li>– <b>последовательность команд, реализующая алгоритм решения задачи</b></li> </ul>
4.	<p>Укажите оператор выбора в языке C ++.</p> <ul style="list-style-type: none"> <li>– CASE</li> <li>– choice</li> <li>– <b>switch ... case</b></li> <li>– default</li> </ul>
5.	<p>Укажите, в котором выражении правильно применяется операция в целочисленных типов (язык C ++).</p> <ul style="list-style-type: none"> <li>– <b>int a = 2, b = 0x24, c; c = a + b% a;</b></li> <li>– short x = 0, y = 2; for (y&gt;! = x) {...};</li> <li>– int c &lt;&gt; 2;</li> <li>– const char mychar = 2; long s = 3; mychar / = s;</li> </ul>
6.	<p>Для чего предназначен оператор continue в языке C ++?</p> <ul style="list-style-type: none"> <li>– <b>Пропускает остаток тела цикла и переходит к следующей итерации.</b> +</li> <li>– Пропускает цикл и переходит к следующему оператору в теле программы.</li> <li>– Определяет условие продолжения цикла.</li> <li>– Продолжает выполнение текущей итерации цикла.</li> </ul>
7.	<p>Значение переменной number не лежит между 3 и 6. Укажите правильный вариант записи данного утверждения на языке C ++.</p> <ul style="list-style-type: none"> <li>– !number &gt; 3 &amp;&amp; number &lt; 6</li> <li>– ! (Number &lt; 3 &amp;&amp; number &lt; 6)</li> <li>– ! (Number &lt; 6    number &gt; 3)</li> <li>– <b>number &lt; 3    number &gt; 6</b></li> </ul>
8.	<p>Укажите верное утверждение.</p> <ul style="list-style-type: none"> <li>– Строки в C + представляются как массивы элементов типа char, заканчивающиеся символом '0'.</li> <li>– <b>Строки в C + представляются как массивы элементов типа char, заканчивающиеся символом '\0'.</b></li> <li>– Строки в C + представляются как массивы элементов типа char, заканчивающиеся символом »</li> <li>– Строки в C + представляются как массивы элементов типа char, заканчивающиеся символом '.</li> </ul>
9.	<p>Что выполняет операция ++ в языке C ++?</p> <p>Уменьшает значение операнда на единицу.  Уменьшает значение операнда на два.  Увеличивает значение операнда на два.  <b>Увеличивает значение операнда на единицу.</b></p>
10.	<p>Как средствами языка C ++ выполняется потоковое чтение из двоичного (бинарного) файла? Название потока a_file.</p> <ul style="list-style-type: none"> <li>– a_file &gt;&gt; z;</li> <li>– a_file.in (z);</li> <li>– z = a_file.read;</li> <li>– <b>a_file.read ((char *) &amp; z, sizeof z);</b></li> </ul>
11.	<p>В каком из вариантов объявлен двумерный массив в C ++?</p>

	<ul style="list-style-type: none"> <li>- <b>int anarray [20] [20];</b></li> <li>- int array [20, 20];</li> <li>- array anarray [20] [20];</li> <li>- char array [20];</li> </ul>
12.	<p>Укажите запись экранированного символа языке C ++.</p> <ul style="list-style-type: none"> <li>- 'F'</li> <li>- «\ 022»</li> <li>- 'Ю'</li> <li>- '\ t'</li> </ul>
13.	<p>Зачем в C ++ используют оператор return?</p> <p>Чтобы задержать работу программы.  Функция, в которой он содержится, завершает свое выполнение и управление возвращается в <b>то место программы, из которого вызывалась данная функция.</b>  Чтобы организовать цикл.  Чтобы ввести в программу новые значения.</p>
14.	<p>В программе на языке C ++ объявлены такие переменные int x, y; Выражение позволяет вычислить остаток от деления этих переменных?</p> <p><b>x% y</b>  x div y  x mod y  x / y</p>
15.	<pre>int m = 5; namespace space1 { int x1 = 1; namespace space2 { int x1 = 3 + ::m + space1::x1; } } int main() { int x3 = space1::space2::x1 * 2; { int x3 = 20; } cout &lt;&lt; x3; return 0; }</pre> <ul style="list-style-type: none"> <li>- <b>18</b></li> <li>- 20</li> <li>- ошибка компиляции</li> <li>- 4</li> </ul>
16.	<p>Чему будет равен результат вычисления выражения: int d=5; bool b = true, c; c = (!b  (d&gt;3));</p> <ul style="list-style-type: none"> <li>- <b>true</b></li> <li>- Ошибка компилятора</li> <li>- false</li> </ul>
17.	<p>Отметьте все правильные утверждения о массивах в языке C++  элементы массива могут быть разных типов</p> <ul style="list-style-type: none"> <li>- <b>все элементы массива должны быть одного типа</b></li> <li>- <b>элементы в памяти расположены рядом</b></li> <li>- элементы могут быть расположены в памяти по одному</li> <li>- <b>элементы всегда нумеруются с нуля</b></li> </ul>
18.	<p>Какой индекс имеет последний элемент массива A?</p> <pre>int A[6];</pre> <p><b>Ответ:5</b></p>
19.	<p>Требуется заполнить массив именно так:  X = [12 9 6 3 0 -3]</p> <p>Какой оператор надо поместить в тело цикла вместо многоточия?</p> <pre>for ( k=0; k&lt;6; k++ ) { ... }</pre> <ul style="list-style-type: none"> <li>- X[k] = k</li> <li>- X[k] = 12 - 2*k</li> <li>- X[k] = 3*k - 12</li> <li>- X[k] = 3*(k + 1) + 9</li> <li>- <b>X[k] = 12 - 3*k</b></li> </ul>
20.	<p>Требуется заполнить массив именно так:  X = [1 2 4 8 16 32]</p> <p>Какой оператор надо поместить в тело цикла вместо многоточия?</p>

	<pre>X[0] = 1; for ( k=1; k&lt;6; k++ ) {     ... } - X[k] = k - X[k] = 2*k - X[k] = X[k-1] + 1 - <b>X[k] = 2*X[k-1]</b> - X[k] = 2*(X[k-1] - 1)</pre>
21.	<p>Какой оператор надо вставить вместо многоточия, чтобы вывести на экран все элементы массива A[N] с четными номерами? В ответе не используйте пробелы.</p> <pre>k = 0; while ( k &lt; N ) {     cout &lt;&lt; A[k] &lt;&lt; " ";     ... } <b>Ответ k++</b></pre>
22.	<p>Задан массив X[N]. Какое условие надо поставить вместо многоточия, чтобы найти сумму положительных элементов массива в переменной S? Вводите ответ без пробелов.</p> <pre>S = 0; for ( k=0; k&lt;N; k++ )     if ( . . . ) S = S + X[k]; <b>Ответ X[k]&gt;0</b></pre>
23.	<p>Задан массив X[N]. Какое условие надо поставить вместо многоточия, чтобы найти количество четных элементов массива в переменной S?</p> <pre>S = 0; for ( j=0; j&lt;N; j++ )     if ( ... ) S++; - S / 2 == 1 - S % 2 == 0 - X[j] % 2 == 1 - <b>X[j] % 2 == 0</b> - X[j] / 2 == 0</pre>
24.	<p>Что будет напечатано в результате выполнения следующего кода?</p> <pre>char x[] = «Hello, world»; char* p = x; p += 4; cout &lt;&lt; *p; - Hello, world - o, world - <b>o</b></pre>
25.	<p>Что будет выведено на экран после выполнения программы?</p> <pre>string s; s = "123"; s = s + "0" + s; s = s + s; cout &lt;&lt; s; <b>Ответ 12301231230123</b></pre>
26.	<pre>string s = "123"; s = s + s[1] + s + s[2]; cout &lt;&lt; s; <b>Ответ 12321233</b></pre>
27.	<p>Что будет выведено на экран после выполнения этой программы?</p> <pre>string s = "12345"; int n; n = s.length() + 3; cout &lt;&lt; n; <b>Ответ 8</b></pre>
28.	<pre>int c = 0, k; for ( k=0; k&lt;S.length(); k++ )     if ( S[k] == '0' )     {         ...     } <b>Ответ c++; или c=c+1;</b></pre>
29.	<p>Сколько единиц будет выведено на экран?</p>

	<pre>string s = "123456"; while ( s.length() &gt; 0 ) {     s.erase ( 0, 2 );     cout &lt;&lt; '1'; } </pre> <p><b>Ответ 3</b></p>
30.	<p>Существует файл «c:\test.dat» в котором записано «Hello World».Каково будет содержимое файла после выполнения кода:</p> <pre>ofstream outfile(«c:\test.dat»,ios::in); if (!outfile) { cout &lt;&lt; «Ошибка создания файла»; return 1; } outfile &lt;&lt; «!!!»; outfile.close(); </pre> <ul style="list-style-type: none"> <li>– <b>!!!o World</b></li> <li>– Hello World!!!</li> <li>– !!!</li> <li>– Является</li> </ul>
31.	<p>С помощью какого метода можно изменить текущую позицию в файле?</p> <ul style="list-style-type: none"> <li>– open</li> <li>– put</li> <li>– <b>seek</b></li> <li>– seaK</li> </ul>
32.	<p>В чем разница между фактическими и формальными параметрами функций?</p> <ul style="list-style-type: none"> <li>– Формальные параметры могут использоваться только вне тела функции, а фактические — используются как вне функции, так и внутри нее.</li> <li>– Нет разницы, это одни и те же параметры.</li> <li>– И формальные, и фактические параметры используются вне тела функции.</li> <li>– <b>Формальные параметры определены в теле функции, а фактические — значение, с которыми функция вызывается.</b></li> </ul>
33.	<p>В каком файле заголовков определен объект cout?</p> <ul style="list-style-type: none"> <li>– stream.h</li> <li>– sysutils.hpp</li> <li>– cout.h</li> <li>– <b>iostream.h</b></li> </ul>
34.	<p>Если в программе уже имеется функция с прототипом int func(int k, double f), то какое из следующих объявлений не вызовет ошибки компиляции?</p> <pre>void func(int m, double g = 3.14) </pre> <p><b>Ответ int func(double x, int y)</b>  <b>double func(int m, double g)</b></p>
35.	<p>Асимптотическая сложность алгоритма это –</p> <ul style="list-style-type: none"> <li>– <b>это функция, позволяющая определить, как быстро увеличивается время работы алгоритма с увеличением объема данных;</b></li> <li>– это функция, позволяющая определить, сколько времени потребуется для решения задачи</li> <li>– это функция, позволяющая определить, сколько действий требуется выполнить алгоритму для решения задачи;</li> <li>– это функция, позволяющая определить, какое количество повторений выполнит алгоритм для решения задачи.</li> </ul>
36.	<p>Смысл динамического программирования –</p> <ul style="list-style-type: none"> <li>– Способ решения сложных задач, в которых для решения обязательно используется динамически выделяемая память;</li> <li>– Это программирование с использованием динамических структур памяти;</li> <li>– Это программирование с использованием динамических массивов;</li> <li>– <b>Способ решения сложных задач путем разбиения их на более простые задачи.</b></li> </ul>
37.	<p>В чем смысл нахождения наибольшей общей последовательности?</p> <ul style="list-style-type: none"> <li>– Это алгоритм нахождения наибольшей неупорядоченной последовательности в строке;</li> <li>– <b>Задача поиска последовательности, которая является подпоследовательностью нескольких последовательностей;</b></li> <li>– Это алгоритм нахождения наибольшей упорядоченной последовательности в строке;</li> </ul>

	– Это тоже самое, что и нахождение подстроки в строке.
38.	Граф – – Только две точки, которые соединяются линиями; – Множество точек, которые могут соединяться линиями; – Множество точек, которые никогда не соединяются линиями; – <b>Множество точек, две из которых обязательно соединяются линиями</b>
39.	О каком алгоритме здесь идет речь? «Если ребро ведет в вершину, которая не была рассмотрена ранее, то запускаем алгоритм от этой нерассмотренной вершины, а после возвращаемся и продолжаем перебирать ребра. Возврат происходит в том случае, если в рассматриваемой вершине не осталось ребер, которые ведут в нерассмотренную вершину. Если после завершения алгоритма не все вершины были рассмотрены, то необходимо запустить алгоритм от одной из рассмотренных вершин»? – <b>Обход графа в глубину;</b> – Обход графа в ширину; – Волновой алгоритм.
40.	Алгоритм сортировки  – Это алгоритм заполнения списка упорядоченными данными; – Это алгоритм удаления из списка неупорядоченных данных; – <b>Это алгоритм для упорядочивания элементов в списке;</b> – Это алгоритм выборки из списка упорядоченных данных.
41.	Как работает сортировка методом выбора: – элементы входной последовательности просматриваются по одному, и каждый новый поступивший элемент размещается в подходящее место среди ранее упорядоченных элементов. – алгоритм проходит по массиву, сравнивая рядом стоящие элементы и меняя их местами. Проходы производятся поочередно, то слева, то справа. Проходы повторяются до тех пор, пока массив не будет отсортирован. – <b>находим номер минимального значения в текущем списке. Производим обмен этого значения со значением первой не отсортированной позиции. Сортируем хвост списка, исключив из рассмотрения уже отсортированные элементы.</b> – алгоритм проходит по массиву, сравнивая рядом стоящие элементы и меняя их местами. Проходы повторяются до тех пор, пока массив не будет отсортирован.
42.	В процессе сортировки сравниваются элементы, отстоящие друг от друга на некоторое, возможно большое, число позиций. По какому алгоритму выполняется эта сортировка?  – пузырьковая; – отбором; – вставками; – быстрая; – <b>Шелла.</b>
43.	В процессе сортировки возможно перемещение по массиву большого числа элементов. По какому алгоритму выполняется эта сортировка?  – отбором; – быстрая; – Шелла; – вставками; – <b>пузырьковая.</b>
44.	Лист дерева-  – это последний первый узел, расположенный на нижнем уровне; – <b>конечный узел, из которого не выходит ни одна дуга;</b> – связь между узлами; – это самый первый узел, расположенный на верхнем уровне.
45.	Корень дерева – это последний первый узел, расположенный на нижнем уровне; – связь между узлами; – <b>это самый первый узел, расположенный на верхнем уровне;</b> – конечный узел из которого не выходит ни одна дуга.
46.	Дек – это – <b>линейный список, в котором можно добавлять и удалять элементы как с одного, так и с другого конца;</b>

	<ul style="list-style-type: none"> <li>– структура данных, которая работает по принципу FILO (First In – Last Out), “первый пришёл – последний ушёл”;</li> <li>– структура данных, которая работает по принципу FIFO (First In – First Out), “первым пришёл – первым ушёл”.</li> </ul>
47.	<p>Очередь – это</p> <ul style="list-style-type: none"> <li>– линейный список, в котором, можно добавлять и удалять элементы как с одного, так и с другого конца;</li> <li>– структура данных, которая работает по принципу FILO (First In – Last Out), “первый пришёл – последний ушёл”;</li> <li>– <b>структура данных, которая работает по принципу FIFO (First In – First Out), “первым пришёл – первым ушёл”</b></li> </ul>
48.	<p>Что более точно отражает структуру дерево –</p> <ul style="list-style-type: none"> <li>– это список двухсвязных элементов;</li> <li>– это структура, состоящая из набора узлов и связей между ними;</li> <li>– это список односвязных элементов;</li> <li>– <b>это структура, отражающая иерархию.</b></li> </ul>
49.	<p>Какие позиции очереди доступны для занесения новых элементов?</p> <ul style="list-style-type: none"> <li>– <b>конец очереди;</b></li> <li>– все позиции;</li> <li>– начало очереди;</li> <li>– начало и конец очереди;</li> <li>– все позиции, кроме начала очереди.</li> </ul>
50.	<p>Точки графа называются</p> <ul style="list-style-type: none"> <li>– пунктами;</li> <li>– <b>вершинами;</b></li> <li>– узлами;</li> <li>– ребрами.</li> </ul>

## 3.2 Домашнее задание

**3.2.1 Шифр и наименование компетенции ОПК-7** Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

№ задания	Текст задания
51.	<p>Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e.</p> $\ln(1-x) = -\sum_{n=1}^{\infty} \frac{x^n}{n} = -\left(x + \frac{x^2}{2} + \frac{x^4}{4} + \dots\right), \quad -1 \leq x < 1.$
52.	<p>– Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e</p> $\ln \frac{x+1}{x-1} = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = 2 \left( \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots \right),$
53.	<p>Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e</p> $\operatorname{arccotg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1} x^{2n+1}}{2n+1} = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} - \dots, \quad  x  \leq 1.$
54.	<p>Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e</p>

	$\ln(x+1) = \sum_0^{\infty} \frac{(-1)^n x^{n+1}}{n+1} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$
55.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $\ln \frac{1+x}{1-x} = 2 \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = 2 \left( x + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right), \quad  x  < 1.$
56.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $\ln \frac{x+1}{x-1} = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = 2 \left( \frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots \right),$
57.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $\operatorname{arctg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots, \quad x > 1.$
58.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $\operatorname{arctg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots, \quad x > 1.$
59.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $\operatorname{arctg} x = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots, \quad  x  \leq 1.$
60.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $e^{-x} = \sum_0^{\infty} \frac{x^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots, \quad x < \infty$
61.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $\operatorname{arccotg} x = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1} x^{2n+1}}{2n+1} = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} + \dots, \quad  x  \leq 1.$
62.	Вычислить и вывести на экран функцию, заданную с помощью ряда Тейлора, на интервале от x1 до x2 с шагом dx с точностью e $e^x = \sum_0^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots, \quad x < \infty$
63.	Подсчитать количество нулей в массиве x. (Составить фрагмент программы, не меняющий значения x, после исполнения которого значение некоторой целой переменной k равнялось бы числу нулей среди компонент массива x.)
64.	Дан массив из n целых чисел, причём x[1] <= x[2] <= ... <= x[n]. Найти количество различных чисел среди элементов этого массива.
65.	Строка разбита на n квадратных клеток. Некоторые клетки покрашены в черный цвет. Считая, что цвета клеток даны в виде массива типа из n логических элементов; подсчитать число черных клеток
66.	Дан массив из n целых чисел. Найти количество различных чисел среди элементов этого массива. (Число действий должно быть порядка n*n.)
67.	Дан массив x из n целых чисел, известно, что все элементы массива - числа от 1 до k. Найти количество различных чисел среди элементов этого массива. Число действий должно быть порядка n+k.
68.	Дан массив целых чисел x[1]..x[m+n], рассматриваемый как соединение двух его отрезков: начала x[1]..x[m] длины m и конца x[m+1]..x[m+n] длины n. Не используя дополнительных массивов, переставить начало и конец. (Число действий порядка m+n.)
69.	Коэффициенты многочлена хранятся в массиве из n целых чисел (n - натуральное число, степень многочлена). Вычислить значение этого многочлена в точке x (т. е. a[n]*(x в степени n)+...+a[1]*x+a[0]).
70.	Дан массив x [1]..x[n] целых чисел. Не используя других массивов, переставить элементы массива в обратном порядке.

71.	Дан массив из $n$ целых чисел. Произвести перестановку элементов в порядке убывания.
72.	Дан массив $x[1]..x[n]$ целых чисел. Не используя других массивов, переставить элементы массива в обратном порядке.
73.	В неупорядоченном массиве целых чисел (как положительных, так и отрицательных) найти сегмент с максимальной суммой элементов.
74.	Подсчитать количество нулей в массиве $x$ . (Составить фрагмент программы, не меняющий значения $x$ , после исполнения которого значение некоторой целой переменной $k$ равнялось бы числу нулей среди компонент массива $x$ .)
75.	Дан массив из $n$ целых чисел, причём $x[1] \leq x[2] \leq \dots \leq x[n]$ . Найти количество различных чисел среди элементов этого массива.
76.	Строка разбита на $n$ квадратных клеток. Некоторые клетки покрашены в черный цвет. Считая, что цвета клеток даны в виде массива $[0..n]$ типа <code>bool</code> ; подсчитать число черных клеток
77.	Дан массив из $n$ целых чисел. Найти количество различных чисел среди элементов этого массива. (Число действий должно быть порядка $n^2$ .)
78.	В неупорядоченном массиве целых чисел (как положительных, так и отрицательных) найти сегмент с максимальной суммой элементов
79.	Подсчитать количество нулей в массиве $x$ . (Составить фрагмент программы, не меняющий значения $x$ , после исполнения которого значение некоторой целой переменной $k$ равнялось бы числу нулей среди компонент массива $x$ .)
80.	Дан массив целых чисел, причём $x[1] \leq x[2] \leq \dots \leq x[n]$ . Найти количество различных чисел среди элементов этого массива.
81.	Строка разбита на $n$ квадратных клеток. Некоторые клетки покрашены в черный цвет. Считая, что цвета клеток даны в виде массива из $n$ <code>bool</code> элементов; подсчитать число черных клеток
82.	Дан массив из $n$ целых элементов. Найти количество различных чисел среди элементов этого массива. (Число действий должно быть порядка $n^2$ .)
83.	Дан массив из $n$ целых элементов, известно, что все элементы массива - числа от 1 до $k$ . Найти количество различных чисел среди элементов этого массива. Число действий должно быть порядка $n+k$ .
84.	Дан массив целых чисел $x[1]..x[m+n]$ , рассматриваемый как соединение двух его отрезков: начала $x[1]..x[m]$ длины $m$ и конца $x[m+1]..x[m+n]$ длины $n$ . Не используя дополнительных массивов, переставить начало и конец. (Число действий порядка $m+n$ .)
85.	Коэффициенты многочлена хранятся в массиве из $n$ целых чисел ( $n$ - натуральное число, степень многочлена). Вычислить значение этого многочлена в точке $x$ (т. е. $a[n] \cdot x^n + \dots + a[1] \cdot x + a[0]$ ).
86.	Напишите программу, которая вводит строку и выводит ее, сокращая каждый раз на символ до тех пор, пока в строке не останется 1 символ.
87.	Напишите программу, определяющую число слов в строке. Одно слово от другого отделяется 1 пробелом.
88.	Составьте программу, определяющую, является ли введенное слово числом.
89.	Введите 2 целых числа. Преобразуйте числа в две строки, объедините их в одну строку и выведите на экран результат.
90.	Напишите программу, которая удаляет из введенной строки любой требуемый введенный с клавиатуры символ.
91.	Составьте программу, удаляющую все пробелы из введенной строки. Примените оператор <code>Repeat</code> и функцию <code>Pos</code> .
92.	Вычислите длину самого короткого слова в предложении из трех слов, разделенное пробелами.
93.	Выясните, какая из букв первая или последняя встречается в заданном слове чаще.
94.	Составьте программу, которая создает файл, состоящий из 10 значений типа <code>int</code> . Прочитайте файл и вычислите сумму элементов.
95.	Составьте программу, которая создает файл, состоящий из 10 значений типа <code>int</code> . Прочитайте файл и выведите элементы в обратном порядке
96.	Напишите программу, которая считывает текст из текстового файла, добавляет к существующему тексту информацию. В программе необходимо проверить наличие текстового файла с заданным именем на диске.
97.	Составьте программу, которая создает файл, состоящий из 15 значений типа <code>int</code> . Прочитайте файл и найдите минимальный элемент.
98.	Составьте программу, которая создает файл о измерении ежедневной температуры, компоненты

	<p>которого имеют следующую структуру.</p> <ul style="list-style-type: none"> <li>• номер измерения (тип int);</li> <li>• значение (тип float).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла.</p>
99.	<p>3. Составьте программу, которая создает файл <i>Vaza</i>, содержащую информацию для студенческой базы данных, компоненты которой имеют следующую структуру:</p> <ul style="list-style-type: none"> <li>• номер студенческого билета (тип int);</li> <li>• ФИО (тип string);</li> <li>• год рождения (тип int);</li> <li>• адрес (тип string).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла</p>
100.	<p>Составьте программу, которая создает файл, содержащий информацию о почтовой базе данных, о подписчиках на газеты и журналы компоненты которого имеют следующую структуру:</p> <ul style="list-style-type: none"> <li>• ФИО (тип string);</li> <li>• адрес (тип string).</li> <li>• название журнала (тип string).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла.</p>
101.	<p>Составьте программу, которая создает файл о измерении ежедневной температуры, компоненты которого имеют следующую структуру.</p> <ul style="list-style-type: none"> <li>• номер измерения (тип int);</li> <li>• значение (тип float).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла</p>
102.	<p>Составьте программу, обеспечивающую создание на диске текстового файла и записи в него текста. Переименуйте файл. Выведите информацию на экран.</p>
103.	<p>Составьте программу, которая создает файл содержащую информацию о хранящемся на складе товаре компоненты которого имеют следующую структуру:</p> <ul style="list-style-type: none"> <li>• код товара (тип int);</li> <li>• наименование товара (тип string);</li> <li>• цену (тип float).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла.</p>
104.	<p>Составьте программу, которая создает файл <i>Voln</i>, содержащий информацию больничной базы данных о стационарных больных, компоненты, которого имеют следующую структуру:</p> <ul style="list-style-type: none"> <li>• ФИО (тип string);</li> <li>• возраст (тип int);</li> <li>• диагноз (тип string);</li> <li>• ФИО лечащего врача (тип string).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла</p>
105.	<p>Составьте программу, которая создает файл содержащий информацию о жильцах дома, отображая в нем следующую информацию о каждом:</p> <ul style="list-style-type: none"> <li>• номер квартиры;</li> <li>• ФИО;</li> <li>• Возраст.</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла.</p>
106.	<p>Составьте программу, которая создает файл, содержащий информацию о хранящемся на складе товаре компоненты которого имеют следующую структуру:</p> <ul style="list-style-type: none"> <li>• код товара (тип int);</li> <li>• наименование товара (тип string);</li> <li>• цену (тип float).</li> </ul> <p>Обеспечьте ввод нескольких значений и вывод всего файла.</p>
107.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> <li>1. Сумму отрицательных элементов массива.</li> <li>2. Произведение элементов массива, расположенных между максимальным и минимальным элементами.</li> </ol> <p>Упорядочить элементы массива по возрастанию.</p>
108.	<p>В одномерном массиве, состоящем из <math>n</math> вещественных элементов, вычислить:</p> <ol style="list-style-type: none"> <li>1. Сумму положительных элементов массива.</li> <li>2. Произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.</li> </ol> <p>Упорядочить элементы массива по убыванию.</p>
109.	<p>В одномерном массиве, состоящем из <math>n</math> целочисленных элементов, вычислить:</p> <ol style="list-style-type: none"> <li>1. Произведение элементов массива с четными номерами.</li> <li>2. Сумму элементов массива, расположенных между первым и последним нулевыми элементами.</li> </ol>

	Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом — все отрицательные (элементы, равные нулю, считать положительными).
110.	В одномерном массиве, состоящем из $n$ вещественных элементов, вычислить: 1. Сумму элементов массива с нечетными номерами. 2. Сумму элементов массива, расположенных между первым и последним отрицательными элементами. Сжать массив, удалив из него все элементы, модуль которых не превышает единицу. Освободившиеся в конце массива элементы заполнить нулями
111.	В одномерном массиве, состоящем из $n$ вещественных элементов, вычислить: 1. Количество элементов массива, лежащих в диапазоне от $A$ до $B$ . 2. Сумму элементов массива, расположенных после максимального элемента. Упорядочить элементы массива по убыванию модулей.
112.	В одномерном массиве, состоящем из $n$ вещественных элементов, вычислить: 1. Количество элементов массива, равных нулю. 2. Сумму элементов массива, расположенных после минимального элемента. Упорядочить элементы массива по возрастанию модулей.
113.	Найти наименьший элемент в упорядоченном массиве $A$ с помощью линейного, бинарного и индексно-последовательного поиска.
114.	Найти элементы в упорядоченном массиве $A$ , которые больше $30$ , с помощью линейного, бинарного и индексно-последовательного поиска.
115.	Вывести на экран все числа массива $A$ кратные $3$ ( $3, 6, 9, \dots$ ) с помощью линейного, бинарного и индексно-последовательного поиска.
116.	Создать копию списка.
117.	Добавить элемент в начало списка.
118.	Склеить два списка.
119.	Удалить $n$ -ый элемент из списка.
120.	Вставить элемент после $n$ -го элемента списка.
121.	Упорядочить элементы в списке по возрастанию.
122.	Удалить каждый второй элемент списка.
123.	Удалить каждый третий элемент списка.
124.	Упорядочить элементы списка по убыванию.
125.	Очистить список
126.	1. Описать процедуру или функцию, которая: а) присваивает параметру $E$ запись из самого левого листа непустого дерева $T$ (лист-вершина, из которого не выходит ни одной ветви); б) определяет число вхождений записи $E$ в дерево $T$ .
127.	2. Вершины дерева вещественные числа. Описать процедуру или функцию, которая: а) вычисляет среднее арифметическое всех вершин дерева; б) добавляет в дерево вершину со значением, вычисленным в предыдущей процедуре (функции).
128.	3. Записи вершин дерева - вещественные числа. Описать процедуру, которая выбирает все вершины с отрицательными записями и строит из них новое дерево.
129.	4. Записи вершин дерева - вещественные числа. Описать процедуру или функцию, которая: а) находит максимальное или минимальное значение записей вершин непустого дерева; б) печатает записи из всех листьев дерева.
130.	5. Описать процедуру или функцию, которая: а) определяет, входит ли вершина с записью $E$ в дерево $T$ ; б) если такая запись не найдена, то она добавляется. 6. Описать процедуру или функцию, которая: а) находит в непустом дереве $T$ длину (число ветвей) пути от корня до ближайшей вершины с записью $E$ ; если $E$ не входит в $T$ , то за ответ принять $-1$ . б) определяет максимальную глубину непустого дерева $T$ , т.е. число ветвей в самом длинном из путей от корня дерева до листьев.

### 3.3 Зачет

#### Вопросы для зачёта

**3.3.1 Шифр и наименование компетенции ОПК-7** Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять

обоснованный выбор инструментария программирования и способов организации программ.

Номер вопроса (задачи, задания)	Текст вопроса (задачи, задания)
131.	Понятия: алгоритм, типы данных, структуры данных.
132.	Классификация структур данных.
133.	Массивы, описание, хранение в памяти, обращение к элементам массива
134.	Матрицы, описание, хранение в памяти, обращение к элементам матрицы.
135.	Строки, описание, обращение к элементам строки, основные операторы обработки строки
136.	Файлы. Понятие файловой переменной.
137.	Работа с файлами в потоке. Открытие файлов для записи.
138.	Работа с файлами в потоке. Открытие файлов для чтения.
139.	Функции. Формальные и фактические параметры
140.	Постановка задачи. Идея динамического программирования. Сравнение рекурсии и динамического программирования
141.	Понятие рекурсии. Пример рекурсивной программы.
142.	Классы сложности алгоритмов.
143.	Алгоритмы обработки данных линейной структуры – сортировка.
144.	Сортировка выбором.
145.	Сортировка обменом (пузырьком).
146.	Сортировка вставками.
147.	Сортировка Шелла.
148.	Быстрая сортировка.
149.	Алгоритмы обработки данных линейной структуры – поиск.
150.	Методы поиска.
151.	Последовательный поиск.
152.	Бинарный поиск.
153.	Односвязные списки. Описание элементов
154.	Очереди. Описание элементов
155.	Бинарные деревья. Способы обхода дерева

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

Процедуры оценивания в ходе изучения дисциплины знаний, умений и навыков, характеризующих этапы формирования компетенций, регламентируются положениями:

- П ВГУИТ 2.4.03 Положение о курсовых экзаменах и зачетах;
- П ВГУИТ 4.1.02 Положение о рейтинговой оценке текущей успеваемости.

Для оценки знаний, умений, навыков обучающихся по дисциплине применяется рейтинговая система. Итоговая оценка по дисциплине определяется на основании определения среднеарифметического значения баллов по каждому заданию.

Зачет по дисциплине выставляется в зачетную ведомость по результатам работы в семестре после выполнения всех видов учебной работы, предусмотренных рабочей программой дисциплины (с отметкой «зачтено») и получении по результатам тестирования по всем разделам дисциплины не менее 60 %.

## 5. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания для каждого результата обучения по дисциплине

Результаты обучения по этапам формирования компетенций	Предмет оценки (продукт или процесс)	Показатель оценивания	Критерии оценивания сформированности компетенций	Шкала оценивания	
				Академическая оценка или баллы	Уровень освоения компетенции
ОПК-7 Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ					
<b>ЗНАТЬ:</b> основные виды структур данных; алгоритмы обработки информации, хранящейся в различных видах структур данных; достоинства и недостатки каждого вида структур данных для применения при решении различных задач.	Тестирование	Правильность ответов при тестировании	Обучающийся ответил на 85-100 % вопросов	85-10 / отлично/зачтено	освоена/повышенный
			Обучающийся ответил на 75-84 % вопросов	75-84/ хорошо/зачтено	освоена/повышенный
			Обучающийся ответил на 60-69 % вопросов	60-74/ удовлетворительно/ зачтено	освоена/базовый
			Обучающийся ответил на 0-59 % вопросов	0-59/ неудовлетворительно/не зачтено	не освоена (недостаточный)
	Зачет собеседование	Правильность логичность ответов собеседования	обучающийся показал глубокие знания программного материала, грамотно и логично его излагает.	зачтено /отлично	освоена/повышенный
			обучающийся твердо знает программный материал, грамотно его излагает, не допускает существенных неточностей в ответе.	зачтено/хорошо	освоена/повышенный
			обучающийся имеет знания только основного материала, но не усвоил его деталей, не допускает грубых ошибок в ответе, требует в отдельных случаях наводящих вопросов для принятия правильного решения.	зачтено/удовлетворительно	освоена/базовый
			обучающийся допускает грубые ошибки в ответе.	не зачтено/ неудовлетворительно	не освоена (недостаточный)
Зачет решение задач по программированию на C++	Правильность логичность лаконичность программного кода	обучающийся выбрал верную методику решения задачи, программный код отвечает требованиям лаконичности, программа, работа оформлена в соответствии со стандартом вуза, без замечаний, верно ответил на все вопросы.	отлично/зачтено	освоена/повышенный	

			обучающийся выбрал верную методику решения задачи, работа оформлена в соответствии со стандартом вуза, ответил на все вопросы, но имеются незначительные замечания по тексту и оформлению работы, допустил не более 2 незначительных ошибок в ответе, которые самостоятельно исправил.	хорошо/зачтено	освоена/повышенный
			обучающийся выбрал верную методику решения задачи, однако имеются замечания по коду программы, имеются замечания по тексту и оформлению работы, допустил не более 3 ошибок в ответе, который исправил с наводящими вопросами, неуверенно владеет приемами работы с ИТ.	удовлетворительно/зачтено	освоена/базовый
			обучающийся не может применять полученные знания на практике, не реализовал программу.	неудовлетворительно/не зачтено	не освоена (недостаточный)
<b>УМЕТЬ:</b> делать обоснованный выбор используемых при решении задач структур данных; применять структуры данных и алгоритмы их обработки при решении различных задач; применять полученные знания и навыки в своей дальнейшей профессиональной деятельности.	Тестирование	правильность ответов на мини задачи по программированию на С++	Обучающийся ответил на 85-100 % вопросов	85-10 / отлично/зачтено	освоена/повышенный
			Обучающийся ответил на 75-84 % вопросов	75-84/ хорошо/зачтено	освоена/повышенный
			Обучающийся ответил на 60-69 % вопросов	60-74/ удовлетворительно/зачтено	освоена/базовый
			Обучающийся ответил на 0-59 % вопросов	0-59/ неудовлетворительно/не зачтено	не освоена (недостаточный)
<b>ВЛАДЕТЬ:</b> навыками создания собственных и использования библиотечных структур данных при разработке программ на языке программирования С++.	Тестирование	Правильность ответов при тестировании	Обучающийся ответил на 85-100 % вопросов	85-10 / отлично/зачтено	освоена/повышенный
			Обучающийся ответил на 75-84 % вопросов	75-84/ хорошо/зачтено	освоена/повышенный
			Обучающийся ответил на 60-69 % вопросов	60-74/ удовлетворительно/зачтено	освоена/базовый
			Обучающийся ответил на 0-59 % вопросов	0-59/ неудовлетворительно/не зачтено	не освоена (недостаточный)

	Домашнее задание	Правильность логичность лаконичность программногo кода	обучающийся выбрал верную методику решения задачи, программный код отвечает требованиям лаконичности, программа, работа оформлена в соответствии со стандартом вуза, без замечаний, верно ответил на все вопросы.	85-100 / отлично/зачтено	освоена/повышенный
			обучающийся выбрал верную методику решения задачи, работа оформлена в соответствии со стандартом вуза, ответил на все вопросы, но имеются незначительные замечания по тексту и оформлению работы, допустил не более 2 незначительных ошибок в ответе, которые самостоятельно исправил.	75-84/ хорошо/зачтено	освоена/повышенный
			обучающийся выбрал верную методику решения задачи, однако имеются замечания по коду программы, имеются замечания по тексту и оформлению работы, допустил не более 3 ошибок в ответе, который исправил с наводящими вопросами, неуверенно владеет приемами работы с ИТ.	60-74/ удовлетворительно/ зачтено	освоена/базовый
			обучающийся не может применять полученные знания на практике, не реализовал программу и .	0-59/ неудовлетворительно/не зачтено	не освоена (недостаточный)