

МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ВГУ»)

Заведующий кафедрой  
Программного обеспечения и администрирования  
информационных систем



Артемов М. А.  
02.04.2024 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**  
**Б1.О.29 Структуры и алгоритмы компьютерной обработки данных**

1. Код и наименование направления подготовки/специальности:  
02.03.03 Математическое обеспечение  
и администрирование информационных систем
2. Профиль подготовки/специализация:  
Проектирование и разработка информационных систем
3. Квалификация выпускника: бакалавр
4. Форма обучения: очная
5. Кафедра, отвечающая за реализацию дисциплины: Математическое обеспечение  
и администрирование информационных систем
6. Составители программы: Каплиева Наталья Алексеевна,  
кандидат физико-математических наук, доцент
7. Рекомендована: НМС факультета ПММ, протокол № 5 от 22.03.2024

---

*отметки о продлении вносятся вручную)*

8. Учебный год: 2024/2025

Семестр(ы): 3, 4

**9. Цели и задачи учебной дисциплины**

Цели дисциплины – формирование у студентов профессиональных компетенций, связанных с использованием знаний в области структур данных и теории алгоритмов, пониманием концепции абстрактных типов данных и подходов к их реализации на языке C++ на основе принципов объектно-ориентированного построения программ, оценки влияния выбора структур данных и/или алгоритмов на эффективность программы.

Основными **задачами** изучения дисциплины являются:

- получение практических навыков решения задач профессиональной деятельности с использованием разных структур данных (стеков, очередей, деревьев, хэш-таблиц), используя концепции абстракции данных и модульного программирования;
- развитие умений, основанных на полученных теоретических знаниях, предлагать и применять эффективные подходы к решению (алгоритмизации) поставленных задач с использованием данных простой и сложной структуры;
- получение навыков самостоятельной работы, предполагающей изучение специфических особенностей работы со структурами данных в рамках разработки подходов (алгоритмов) к решению поставленной задачи, вопросов управления памятью в C++ и использования компонентов стандартной библиотеки шаблонов (STL).

**10. Место учебной дисциплины в структуре ООП:** Дисциплина «Структуры и алгоритмы компьютерной обработки данных» входит в основную часть программы бакалавриата (Б1). Изучение данного курса должно базироваться на знаниях обучающимся материала курса «Информатика и программирование». Дисциплина является базовой для изучения курсов: «Численные методы», «Алгоритмы компьютерной графики», «Программирование на C#», «Программирование встроенных систем».

**11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями) и индикаторами их достижения:**

Код	Название компетенции	Код(ы)	Индикатор(ы)	Планируемые результаты обучения
ОПК-3	Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	ОПК-3.1	Демонстрирует знание современных информационных технологий и применяет их при создании программных продуктов	Знать: базовые абстрактные типы (структуры) данных (контейнеры), понимать их особенности, применяемые операции и методы реализации на языке C++; контейнеры, итераторы и алгоритмы как основные компоненты STL, приемы хэширования и разрешения коллизий.  Уметь: разрабатывать программы на C++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.
		ОПК-3.2	Реализует различные новые программные продукты с использованием современных информационных технологий	Владеть: навыками процедурного и объектно-ориентированного программирования на C++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.

**12. Объем дисциплины в зачетных единицах/час.**(в соответствии с учебным планом) — 10/360.

**Форма промежуточной аттестации**(зачет/экзамен) зачет, экзамен

**13. Трудоемкость по видам учебной работы**

Вид учебной работы		Трудоемкость		
		Всего	По семестрам	
			3 семестр	4 семестр
Контактная работа		160	80	80
в том числе:	лекции	64	32	32

	практические	32	16	16
	лабораторные	64	32	32
	курсовая работа			
Самостоятельная работа		128	64	64
Промежуточная аттестация (для экзамена)		72	36	36
Итого:		360	180	180

### 13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК *
<b>1. Лекции</b>			
1.1	Стеки	Понятие стека. Методы работы со стеком	
1.2	Очереди	Понятие очереди. Методы работы с очередью	
1.3	Бинарные деревья	Понятие бинарного дерева. Сбалансированные деревья, деревья поиска. Рекурсивная и некурсивная обработки бинарного дерева	
1.4	Сильноветвящиеся деревья	Способы представления сильноветвящихся деревьев. В-деревья. Trie-деревья	
1.5	Объектно-ориентированное программирование	Принципы, заложенные в основу объектно-ориентированной-технологии. Объектно-ориентированная декомпозиция. Механизмы разработки классов; наследование, композиция, наполнение, полиморфизм. Механизмы определения типов объектов на этапе выполнения программы.	
1.6	STL-контейнеры	STL-контейнеры. Способы обработки элементов контейнера	
1.7	Сортировки	Внутренние сортировки. Внешние сортировки.	
1.8	Хеширование	Хеширование как метод преобразование массива входных данных произвольной длины в (выходную) битовую строку установленной длины	
1.9	Алгоритмы с возвратом	Поиск одного решения, поиск всех решений, поиск оптимального решения	
1.10	Основы параллельного программирования	Понятия процесса и потока. Независимые потоки, способы возврата результатов. Зависимые потоки, способы работы с разделяемыми ресурсами	
<b>2. Практические занятия</b>			
2.1	Стеки	Методы работы со стеком	
2.2	Очереди	Методы работы с очередью	
2.3	Бинарные деревья	Сбалансированные деревья, деревья поиска (добавление и удаление элементов), дерево-формула	
2.4	Сильноветвящиеся деревья	Trie-деревья. Алгоритмы добавления, удаления, поиска	
2.5	Объектно-ориентированное программирование	Объектно-ориентированная декомпозиция. Механизмы разработки классов; наследование, композиция, наполнение, полиморфизм. Механизмы определения типов объектов на этапе выполнения программы.	
2.6	STL-контейнеры	STL-контейнеры. Способы обработки элементов контейнера	
2.7	Сортировки	Внутренние сортировки. Внешние сортировки.	
2.8	Хеширование	Хэш-функция, ее свойства. Способы разрешения коллизии	
2.9	Алгоритмы с возвратом	Поиск одного решения, поиск всех решений, поиск оптимального решения	

2.10	Основы параллельного программирования	Асинхронные задачи, независимые потоки, способы возврата результатов. Зависимые потоки, Interlocked-функции, атомарные типы, сигнальные объекты ядра
<b>3. Лабораторные работы</b>		
3.1	Стеки	Методы работы со стеком
3.2	Очереди	Методы работы с очередью
3.3	Бинарные деревья	Сбалансированные деревья, деревья поиска, древо-формула. Рекурсивная и нерекурсивная обработки деревьев
3.4	Сильноветвящиеся деревья	Trie-деревья. Алгоритмы добавления, удаления, поиска.
3.5	Объектно-ориентированное программирование	Объектно-ориентированная декомпозиция. Механизмы разработки классов; наследование, композиция, наполнение, полиморфизм. Механизмы определения типов объектов на этапе выполнения программы.
3.6	STL-контейнеры	STL-контейнеры. Способы обработки элементов контейнера
3.7	Сортировки	Внутренние сортировки. Внешние сортировки.
3.8	Хеширование	Хэш-функция, ее свойства. Способы разрешения коллизии
3.9	Алгоритмы с возвратом	Поиск одного решения, поиск всех решений, поиск оптимального решения
3.10	Основы параллельного программирования	Асинхронные задачи, независимые потоки, способы возврата результатов. Зависимые потоки, Interlocked-функции, атомарные типы, сигнальные объекты ядра

### 13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (количество часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Стеки	4	2	2	6	22
2	Очереди	4	2	2	6	22
3	Бинарные деревья	6	2	6	12	36
4	Сильноветвящиеся деревья	6	4	6	14	38
5	STL-контейнеры	4	2	6	10	22
6	Объектно-ориентированное программирование	6	6	8	14	
7	Сортировки	6	4	8	12	32
8	Хеширование	6	2	4	8	24
9	Алгоритмы с возвратом	8	4	8	18	36
10	Основы параллельного программирования	14	4	14	28	56
	Итого:	64	32	64	128	288

### 14. Методические указания для обучающихся по освоению дисциплины

Работа с конспектами лекций, выполнение практических и лабораторных заданий, заданий текущей и промежуточной аттестаций.

При использовании дистанционных образовательных технологий и электронного обучения выполнять все указания преподавателей по работе на LMS-платформе, своевременно подключаться к online-занятиям, соблюдать рекомендации по организации самостоятельной работы

### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины (список литературы оформляется в соответствии с требованиями ГОСТ и используется общая сквозная нумерация для всех видов источников)

а) основная литература:

№ п/п	Источник
1	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: <a href="https://e.lanbook.com/book/156929">https://e.lanbook.com/book/156929</a>
2	Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/119661">https://e.lanbook.com/book/119661</a>
3	Свердлов, С. З. Языки программирования и методы трансляции : учебное пособие / С. З. Свердлов. — 2-е изд., испр. — Санкт-Петербург : Лань, 2019. — 564 с. — ISBN 978-5-8114-3457-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/116391">https://e.lanbook.com/book/116391</a>

б) дополнительная литература:

№ п/п	Источник
4	Тюкачев, Н. А. С#. Алгоритмы и структуры данных : учебное пособие / Н. А. Тюкачев, В. Г. Хлебостроев. — 3-е изд., стер. — Санкт-Петербург : Лань, 2018. — 232 с. — ISBN 978-5-8114-2566-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <a href="https://e.lanbook.com/book/104961">https://e.lanbook.com/book/104961</a> .
5	Бабушкина И.А. Практикум по объектно-ориентированному программированию / И.А. Бабушкина, С.М. Окулов. – М.: "Лаборатория знаний", 2012. – 366 с. Режим доступа: <a href="http://e.lanbook.com/books/element.php?pl1_id=8781">http://e.lanbook.com/books/element.php?pl1_id=8781</a>
6	Липпман С.Б. Язык программирования С++. Базовый курс / С.Б. Липпман., Ж. Лажойе, Б.Э. Му. – М. : Вильямс, 2014. – 1120 с.
7	Пахомов Б.И. С/С++ и MS Visual С++ 2010 для начинающих / Б.И. Пахомов. – СПб. ; БХВ-Петербург, 2012. – 736 с.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
8	<a href="http://www.lib.vsu.ru">www.lib.vsu.ru</a> – ЗНБ ВГУ
9	Окулов С.М. Программирование в алгоритмах. – М. : "Лаборатория знаний", 2014. – 384 с. – Режим доступа: <a href="http://e.lanbook.com/books/element.php?pl1_id=50562">http://e.lanbook.com/books/element.php?pl1_id=50562</a>
10	Онлайн-курс «Языки и методы программирования_ФИИТ». – <a href="https://edu.vsu.ru/course/view.php?id=10368">https://edu.vsu.ru/course/view.php?id=10368</a>

\* Вначале указываются ЭБС, с которыми имеются договора у ВГУ, затем открытые электронно-образовательные ресурсы

**16. Перечень учебно-методического обеспечения для самостоятельной работы (учебно-методические рекомендации, пособия, задачки, методические указания по выполнению практических (контрольных) работ и др.)**

№ п/п	Источник
1	Ускова, О. Ф. Информатика и программирование. Задачник-практикум по структурному программированию на языке С++ / О.Ф. Ускова, Н. А. Каплиева, О. Д. Горбенко. — Воронеж : Издательский дом ВГУ, 2020. — 279 с.
2	Каплиева Н.А. Процедурное программирование : учебно-методическое пособие / Н. А. Каплиева, С. Ю. Болотова. — Воронеж : Издательский дом ВГУ, 2024. — 56 с.
3	Каплиева Н.А. Лямбда-функции. Динамические структуры данных : учебно-методическое пособие / Н. А. Каплиева, С. Ю. Болотова. — Воронеж : Издательский дом ВГУ, 2023. — 56 с.

**17. Образовательные технологии, используемые при реализации учебной дисциплины, включая дистанционные образовательные технологии (ДОТ), электронное обучение (ЭО), смешанное обучение):**

При реализации дисциплины используются модульно-рейтинговая и личностно-ориентированные технологии обучения (ориентированные на индивидуальность студента, компьютерные и коммуникационные технологии). В рамках дисциплины предусмотрены следующие виды лекций: информационная, лекция-визуализация, лекция с применением обратной связи.

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий, для организации самостоятельной работы обучающихся используется онлайн-курс, размещенный на платформе Электронного университета ВГУ (LMS moodle), а также другие Интернет-ресурсы, приведенные в п.15в

### 18. Материально-техническое обеспечение дисциплины:

Лекционная аудитория должна быть оборудована учебной мебелью, компьютером, мультимедийным оборудованием (проектор, экран, средства звуковоспроизведения), допускается переносное оборудование.

Лабораторные занятия должны проводиться в специализированной аудитории, оснащенной учебной мебелью и персональными компьютерами с доступом в сеть Интернет (компьютерные классы, студии), мультимедийным оборудованием (мультимедийный проектор, экран, средства звуковоспроизведения). Число рабочих мест в аудитории должно быть таким, чтобы обеспечивалась индивидуальная работа студента на отдельном персональном компьютере.

Для самостоятельной работы необходимы компьютерные классы, помещения, оснащенные компьютерами с доступом к сети Интернет.

Программное обеспечение (см. файл МТО):

- ОС Windows 8 (10)
- Интернет-браузер (Google Chrome, Mozilla Firefox)
- Microsoft Visual Studio Community Edition (свободное и/или бесплатное ПО)
- Adobe Reader (свободное и/или бесплатное ПО)

### 19. Оценочные средства для проведения текущей и промежуточной аттестаций

Порядок оценки освоения обучающимися учебного материала определяется содержанием следующих разделов дисциплины:

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Модульное программирование	ОПК-3	ОПК-3.1	КИМы (для проведения текущей аттестации) Задания для контрольных работ Задания для лабораторных работ
2	Динамические структуры данных	ОПК-3	ОПК-3.1, ОПК-3.2	
3	Стеки	ОПК-3	ОПК-3.1, ОПК-3.2	
4	Очереди	ОПК-3	ОПК-3.1, ОПК-3.2	
5	Бинарные деревья	ОПК-3	ОПК-3.1, ОПК-3.2	
6	Сильноветвящиеся деревья	ОПК-3	ОПК-3.1, ОПК-3.2	
7	Объектно-ориентированное программирование	ОПК-3	ОПК-3.1, ОПК-3.2	
8	STL-контейнеры	ОПК-3	ОПК-3.1, ОПК-3.2	
9	Сортировки	ОПК-3	ОПК-3.1, ОПК-3.2	
10	Хеширование	ОПК-3	ОПК-3.1, ОПК-3.2	
11	Алгоритмы с возвратом	ОПК-3	ОПК-3.1, ОПК-3.2	
12	Основы параллельного программирования	ОПК-3	ОПК-3.1, ОПК-3.2	

№ п/п	Наименование раздела дисциплины (модуля)	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
	Промежуточная аттестация форма контроля – экзамен			КИМы для проведения промежуточной аттестации

\* В графе «ФОС» в обязательном порядке перечисляются оценочные средства текущей и промежуточной аттестаций.

## 20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущая аттестация проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущая аттестация проводится в формах: устного опроса; защиты лабораторных работ, выполнения контрольных работ.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования. Промежуточная аттестация по итогам освоения дисциплины проводится в форме зачета с оценкой и экзамена. Для получения положительной итоговой оценки необходимо выполнение всех лабораторных и контрольных работ.

### 20.1 Текущий контроль успеваемости

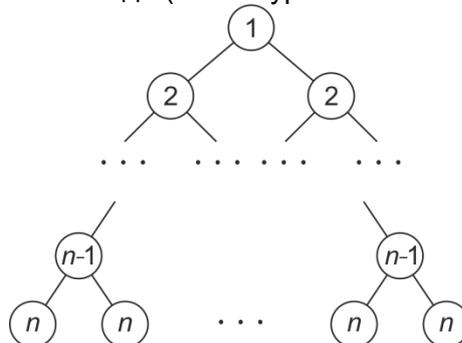
Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств.

Текущая аттестация проводится на занятии одновременно во всей учебной группе в виде теста в электронной образовательной среде «Электронный университет ВГУ», адрес курса — **Курс** «Структуры и алгоритмы компьютерной обработки данных\_МОАИС» <https://edu.vsu.ru/course/view.php?id=10368>, адрес теста текущей аттестации — <https://edu.vsu.ru/mod/assign/view.php?id=1186371> Тест составляется из материалов ФОСа, формируется системой автоматически путём добавления случайных вопросов, количество которых соответствует имеющимся образцам билетов. Большая часть вопросов проверяется автоматически, проверки преподавателем с ручным оцениванием требуют только отдельные вопросы, представленные в форме эссе. Ограничение по времени на каждую попытку — 1 час 30 минут

### Примеры контрольных работ

#### Примеры заданий по теме «Бинарные деревья»

1. Построить дерево заданного вида ( $n$  – натуральное число):

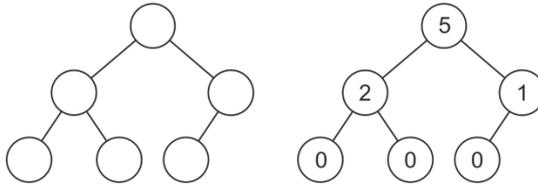


2. Найти сумму элементов на  $n$ -м уровне.

3. Найти глубину дерева.

4. Посчитать количество листьев, имеющих четное значение. Использовать нерекурсивную функцию.

5. Построить копию дерева, в узлах которой записать количество потомков у соответствующего узла в исходном дереве. Пример исходного дерева и его копии представлен на рисунке.



## Примеры лабораторных работ

### Примеры заданий по теме «STL-контейнеры»



Разработать контейнерный класс «Библиотека», в котором список изданий упорядочен по возрастанию значений поля Издательство. Найти издательство, которое чаще всего публиковало журналы в заданном году.

Требования к выполнению заданий (или шкалы и критерии оценивания)

Задание считается выполненным, если разработанный программный продукт полностью решает поставленную задачу, решение предусматривает обработку исключительных ситуаций, обладает удобным пользовательским интерфейсом, соответствует требованиям оформления кода (структурированность, наличие комментариев).

Требования к выполнению заданий текущей и промежуточной аттестаций, шкалы и критерии оценивания

## 20.2 Итоговый контроль успеваемости

Промежуточная аттестация по дисциплине осуществляется с помощью следующих оценочных средств: вопросы к экзамену.

### Вопросы к экзамену

1. Стеки, Очереди. Реализация библиотеки Стек/Очередь: на основе линейного списка; одномерного массива.
2. Бинарные деревья.
3. Trie-деревья.
4. Объектно-ориентированное программирование. Конструкторы. Деструкторы. Наследование. Полиморфизм. Механизм определения типа на этапе выполнения программы.
5. STL-контейнеры. Vector. List. Set. Map.
6. Внутренние сортировки.
7. Внешние сортировки.
8. Хэширование. Хэш-функция. Способы разрешения коллизий.
9. Алгоритмы с возвратом.
10. Основы параллельного программирования. Создание потоков. Независимые потоки. Взаимодействующие потоки. Механизмы синхронизации потоков.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<p>Сформированные знания базовых абстрактных типов (структуры) данных (контейнеры), понимание их особенностей, применяемых операции и методы реализации на языке С++.</p> <p>Сформированные умения разрабатывать программы на С++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.</p> <p>Сформированные навыки процедурного и объектно-ориентированного программирования на С++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.</p>	<i>Повышенный уровень</i>	<i>Отлично</i>
<p>Сформированные, но содержащие отдельные пробелы, знания базовых абстрактных типов (структуры) данных (контейнеры), понимание их особенностей, применяемых операции и методы реализации на языке С++.</p> <p>Сформированные, но содержащие отдельные пробелы, умения разрабатывать программы на С++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.</p> <p>Сформированные, но содержащие отдельные пробелы, навыки процедурного и объектно-ориентированного программирования на С++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.</p>	<i>Базовый уровень</i>	<i>Хорошо</i>
<p>Неполное представление о базовых абстрактных типах (структуры) данных (контейнеры), понимании их особенностей, применяемых операциях и методах реализации на языке С++.</p> <p>Успешное, но не системное умение разрабатывать программы на С++, реализующие заданный алгоритм и использующий определенные структуры данных; разрабатывать хэш-функции и реализовывать способы разрешения коллизий.</p> <p>Неполное владение навыками процедурного и объектно-ориентированного программирования на С++, работы с шаблонами функций и классов, использования STL (контейнеров и алгоритмов), хэширования.</p>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
<p>Фрагментарные знания или отсутствие знаний. Фрагментарные умения или отсутствие умений. Фрагментарные навыки или отсутствие навыков</p>	–	<i>Неудовлетворительно</i>

### **20.3 Фонд оценочных средств сформированности компетенций студентов, рекомендуемый для проведения диагностических работ**

ОПК-3 Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения

ОПК-3.1 Демонстрирует знание современных информационных технологий и применяет их при создании программных продуктов

ОПК-3.2 Реализует различные новые программные продукты с использованием современных информационных технологий

## Вопросы с вариантами ответов (закрытые)

1. Пирамидальная сортировка использует бинарное сортирующее дерево (пирамиду). Выберите вариант массива, образующего пирамиду для сортировки по убыванию.

- A. 2 8 4 3 9 5 1 7 6
- B. 1 3 2 6 9 5 4 7 8
- C. 1 8 2 6 5 4 9 7 3

Ответ: B

2. Коллекция, в которой доступен только последний добавленный элемент. Добавление новых элементов и удаление существующих производится с одного конца.

- A. стек
- B. очередь
- C. хэш-таблица

Ответ: A

3. Коллекция, в которой доступен только первый элемент. Добавление новых элементов производится в конец коллекции, а извлечение из начала.

- A. стек
- B. очередь
- C. хэш-таблица

Ответ: B

4. Коллекция, в которой элементы размещаются по заранее вычисленному ключу. Обеспечивает быстрый поиск элементов.

- A. стек
- B. очередь
- C. хэш-таблица

Ответ: C

5. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренним компонентам:

- A. Статическая
- B. Дружественная
- C. Шаблонная
- D. Полиморфная

Ответ: B

6. Выберите правильное утверждение для абстрактного класса для языка C++:

- A. Класс, у которого все методы чисто виртуальные.
- B. Невозможно создать объект абстрактного класса
- C. В абстрактном классе нельзя описывать методы

Ответ: B

7. С помощью чего реализуется принцип полиморфизма в C++?

- A. Наличие множественного наследования.
- B. Наличие виртуальных методов.
- C. Использование виртуального наследования.
- D. наличие абстрактных классов.

Ответ: B

8. Принцип объектно-ориентированного программирования, заключающийся в объединении атрибутов и методов объекта с целью обеспечения сохранности данных:

- A. Наследование.
- B. Сочетание.
- C. Инициализация.
- D. Инкапсуляция

Ответ: D

9. Данные-компоненты класса инициализировать в теле класса

- A. никогда нельзя
- B. всегда можно
- C. можно все, кроме статических полей
- D. нельзя за исключением статических полей класса

Ответ: D

10. Действие спецификаторов доступа на дружественные функции

- A. не распространяется
- B. распространяется только для спецификатора public
- C. распространяется только для спецификатора private
- D. распространяется только для спецификатора protected

Ответ: A

11. Доступ к внутренним и защищенным полям и функциям

- A. невозможен
- B. возможен только из компонентных и «дружественных» функций
- C. возможен только из компонентных функций
- D. возможен только из «дружественных» функций

Ответ: B

12. Какой из перечисленных методов может быть конструктором для класса String в языке C ++?

- A. String \* String ();
- B. void String ();
- C. String (String & s);
- D. const String (int a);

Ответ: C

13. В программе описаны класс A и объект obj

```
class A
{
public: int a, b, c;
};
A * obj;
```

Как обратиться к атрибуту c?

- A. obj.c
- B. obj->c
- C. obj A ->->c
- D. obj->A.c

Ответ: B

14. Вызовет ли данный код ошибку компиляции?

```
class Figure
{
public:
    int a, b;
    int sum();
    int square();
    ~fig();
};
```

- A. Ошибки нет, все верно.
- B. Ошибка: имя деструктора должно совпадать с именем класса.
- C. Ошибка: имя деструктора не может начинаться с маленькой буквы.
- D. Ошибка: никакой идентификатор в C++ не может начинаться со знака «~».

Ответ: B

15. Укажите правильное объявление виртуального метода, который принимает одно целочисленное значение и возвращает void.

- A. `virtual void SomeFunction(int x);`
- B. `void SomeFunction(int x) virtual;`
- C. `virtual SomeFunction(int x);`
- D. `virtual void SomeFunction(int * x);`

Ответ: A

16. В конструкторах может использоваться список инициализации, который отделяется от заголовка конструктора

- A. двоеточием
- B. знаком равенства
- C. точкой
- D. знаком `->`

Ответ: A

17. В случае, когда один объект передает сообщение другому, говорят, что эти объекты находятся в отношении

- A. использования
- B. включения
- C. диалога
- D. монолога

Ответ: A

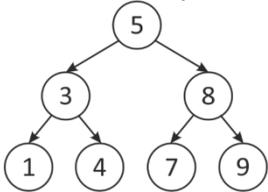
18. Взаимодействие программных объектов в объектно-ориентированной системе осуществляется

- A. посредством передачи управления
- B. путем передачи сообщений
- C. последовательной обработкой данных
- D. за счёт связи объектов

Ответ: B

## Вопросы с кратким текстовым ответом (открытые)

1. Дано бинарное дерево:

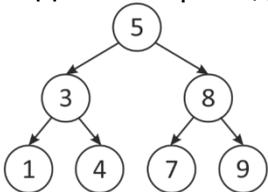


Запишите последовательность, которая будет получена в результате вызова функции (NODE – структура, описывающая узел дерева):

```
void Print(NODE* root)
{
    if (root)
    {
        Print(root->right);
        std::cout << root->info;
        Print(root->left);
    }
}
```

Ответ: 9875431

2. Дано бинарное дерево:

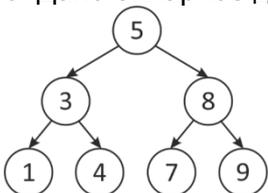


Запишите последовательность, которая будет получена в результате вызова функции (NODE – структура, описывающая узел дерева):

```
void Print(NODE* root)
{
    if (root)
    {
        std::cout << root->info;
        Print(root->left);
        Print(root->right);
    }
}
```

Ответ: 5314879

3. Дано бинарное дерево:



Запишите последовательность, которая будет получена в результате вызова функции (NODE – структура, описывающая узел дерева):

```
void Print(NODE* root)
{
    if (root)
    {
        Print(root->left);
        Print(root->right);
    }
}
```

```
        std::cout << root->info;
    }
}
```

Ответ: 1437985

4. Служебное слово, с помощью которого описывается секция для объявления компонентов, доступных только компонентным функциям того же класса и функциям, объявленным дружественными.

Ответ: private

5. Служебное слово, с помощью которого описывается секция для объявления компонентов, доступных компонентным функциям не только данного класса, но и его потомков.

Ответ: protected

6. Служебное слово, с помощью которого описывается секция для объявления компонентов, доступных за пределами класса в любом месте программы.

Ответ: public

7. Отношение между классами, обеспечивающее возможность конструирования новых более сложных классов из уже имеющихся посредством добавления полей и определения новых методов.

Ответ: Наследование

8. Отношение между классами, при котором объекты одного класса являются неотъемлемой частью другого.

Ответ: Композиция

9. Отношение между классами, при котором количество объектов некоторого класса, включаемых в другой класс, не ограничено и может меняться от нуля до достаточно больших значений.

Ответ: Наполнение

10. Метод класса, который автоматически вызывается при выделении памяти под объект.

Ответ: Конструктор

11. Метод класса, который автоматически вызывается при освобождении объектом памяти.

Ответ: Деструктор

12. Совокупность существенных характеристик некоторого объекта, которые отличают его от всех других видов объектов и, таким образом, четко определяют особенности данного объекта с точки зрения дальнейшего рассмотрения и анализа решаемой задачи.

Ответ: Абстракция

13. Спецификатор, с помощью которого описывается функция, которая, не являясь компонентом некоторого класса, имеет доступ ко всем его компонентам, в том числе внутренним.

Ответ: friend

## **Критерии и шкалы оценивания заданий ФОС:**

Для оценивания выполнения заданий используется балльная шкала:

1) закрытые задания (тестовые с вариантами ответов, средний уровень сложности):

- 1 балл – указан верный ответ;
- 0 баллов – указан неверный ответ (полностью или частично неверный).

2) открытые задания (тестовые с кратким текстовым ответом, повышенный уровень сложности):

- 2 балла – указан верный ответ;
- 0 баллов – указан неверный ответ (полностью или частично неверный).

**Задания раздела 20.3 рекомендуются к использованию при проведении диагностических работ с целью оценки остаточных результатов освоения данной дисциплины (знаний, умений, навыков).**