

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

**УТВЕРЖДАЮ**

Заведующий кафедрой программного обеспечения  
и администрирования информационных систем



Артемов М. А.

08.06.2018 г.

**РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ**

Б1.Б.20 Языки и системы программирования

**1. Шифр и наименование направления подготовки:**

02.03.03 Математическое обеспечение и администрирование информационных систем

**2. Профиль подготовки:** Информационные системы и базы данных

**3. Квалификация (степень) выпускника:** бакалавр

**4. Форма обучения:** очная

**5. Кафедра, отвечающая за реализацию дисциплины:**

Программного обеспечения и администрирования информационных систем

**6. Составители программы:**

Ефремов Максим Сергеевич, к.ф.-м.н., преп.

**7. Рекомендована:** НМС факультета ПММ протокол № 10 от 18.06.2018 г.

**8. Учебный год:** 2018/2019

**Семестр(ы):** 1, 2, 3, 4

**9. Цели и задачи учебной дисциплины:**

Цель курса - приобретение знаний и навыков в области технологии и практики программирования на языке C/C++, формирование культуры разработки программных продуктов на современных языках программирования высокого уровня.

**10. Место учебной дисциплины в структуре ООП:**

Дисциплина входит в базовую часть учебного плана и изучается в 1-м, 2-м, 3-м и 4-м семестрах.

Дисциплина является базовой, поскольку это дисциплина, которая знакомит студентов с современными фундаментальными языками программирования, которые широко используются во всех областях сферы информационных технологий.

### 11. Компетенции обучающегося, формируемые в результате освоения дисциплины:

Компетенция		Планируемые результаты обучения
Код	Название	
ОПК-3	способность использовать и применять углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий	<p>Знать:</p> <p>основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.</p> <p>Уметь:</p> <p>проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.</p> <p>Владеть:</p> <p>навыками практического программирования конкретных задач с использованием языка программирования C/C++.</p>
ОПК-7	способность использовать знания основных концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	<p>Знать:</p> <p>основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.</p> <p>Уметь:</p> <p>проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.</p> <p>Владеть:</p> <p>навыками практического программирования конкретных задач с использованием языка программирования C/C++.</p>
ОПК-8	способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание ПО	<p>Знать:</p> <p>основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.</p> <p>Уметь:</p> <p>проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.</p> <p>Владеть:</p> <p>навыками практического программирования конкретных задач с использованием языка программирования C/C++.</p>
ОПК-11	готовность использовать навыки выбора, проектирования, реализации, оценки	<p>Знать:</p> <p>основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-</p>

	качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки. Уметь: проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки. Владеть: навыками практического программирования конкретных задач с использованием языка программирования C/C++.
--	---	---

## 12. Объем дисциплины в зачетных единицах/часах в соответствии с учебным планом — 11/396.

Форма промежуточной аттестации: зачет с оценкой, экзамен.

### 13. Виды учебной работы

Вид учебной работы	Трудоемкость (часы)				
	Всего	По семестрам			
		1	2	3	4
Аудиторные занятия	144	16	46	50	32
в том числе: лекции	50			34	16
лабораторные	94	16	46	16	16
практические					
Самостоятельная работа	180	20	62	58	40
Итого	324	36	108	108	72
Контроль:	72			36	36
<b>Итого:</b>	<b>396</b>	<b>36</b>	<b>108</b>	<b>144</b>	<b>108</b>
Форма промежуточной аттестации			зачет с оценкой	экзамен	экзамен

#### 13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
<b>1. Лекции</b>		
1.1	Введение. Ввод и вывод.	Обзор возможностей языка C++. Базовые средства вводавывода. Простейшие программы, комментарии.
1.2	Типы и объявления.	Типы, логические значения, символы, целые, типы с плавающей запятой, литералы, void, перечисления, объявления, имена, инициализация, область видимости, объекты, typedef.
1.3	Указатели, массивы и структуры.	Указатели, нулевой указатель, массивы, указатели на массивы, строки, строковые литералы, ссылки, void*, структуры, операторы new и delete.

1.4	Выражения и инструкции.	Обзор операций и операторов, логические операторы, инкремент и декремент, преобразования типов, инструкции, объявления, условные инструкции, циклы.
1.5	Функции.	Объявления и определения функций, вызов функций, передача аргументов по значению и по ссылке, возврат значений, перегрузка функций, правила разрешения неоднозначностей, аргументы по умолчанию, функции с переменным числом аргументов, указатели на функции.
1.6	Исходные файлы. Компиляция.	Компиляция, компоновка, заголовочные файлы и файлы реализации, препроцессор, библиотечные заголовочные файлы, правило одного определения, использование заголовочных файлов для выражения модульности.
1.7	Классы.	Классы, члены класса, управление доступом, конструкторы (по умолчанию, копирования, с аргументами), статические члены, константные члены, деструкторы, объекты классов, выделение памяти.
1.8	Перегрузка операций.	Функции-операции, бинарные и унарные операции, операции - члены и операции - глобальные функции, смешанная арифметика, присваивание, инициализация, операции преобразования типов, операции вызова функции, операция индексации, операции инкремента и декремента.
1.9	Наследование классов. Виртуальные функции.	Производные классы, конструирование и уничтожение классов, иерархии классов, поля типа, виртуальные функции, абстрактные классы, множественное наследование, виртуальное наследование, режимы доступа, способы наследования, статическое и динамическое приведение типов, typeid, информация о типах времени выполнения (RTTI).
1.10	Шаблоны.	Шаблоны, конкретизация, параметры шаблона, шаблоны функций, вывод типов аргументов шаблона, задание аргументов шаблона, перегрузка шаблонов функций, аргументы шаблона по умолчанию, шаблоны классов, наследование и шаблоны.
1.11	Пространства имен. Обработка исключений.	Пространства имен, using, using namespace, разрешение конфликтов имен, поиск имен. Исключения, throw и catch, обработка исключений, группировка исключений, перехват исключений, перехват всех исключений (...), исключения в конструкторах и деструкторах, стандартные исключения.
1.12	Стандартная библиотека.	Контейнеры и алгоритмы стандартной библиотеки, базовые контейнеры, итераторы, доступ к элементам, конструкторы, модификаторы, стандартные контейнеры, стандартные алгоритмы, функторы, предикаты.
<b>2. Лабораторные занятия</b>		
2.1	Введение. Ввод и вывод.	Изучение темы. Разбор примеров, решение и разбор простейших задач.
2.2	Типы и объявления.	Изучение темы. Разбор примеров, решение и разбор простейших задач.

2.3	Указатели, массивы и структуры.	Изучение темы. Разбор примеров, решение и разбор простейших задач. Контрольная работа №1.
2.4	Выражения и инструкции.	Изучение темы. Разбор примеров, решение задач.
2.5	Функции.	Разбор примеров, решение задач. Контрольная работа №2.
2.6	Исходные файлы. Компиляция.	Практическое изучения процесса построения приложения из исходных кодов.
2.7	Классы.	Разбор примеров, решение задач. Контрольная работа №3.
2.8	Перегрузка операций.	Разбор примеров, решение задач.
2.9	Наследование классов. Виртуальные функции.	Разбор примеров, решение задач.
2.10	Шаблоны.	Разбор примеров, решение задач.
2.11	Пространства имен. Обработка исключений.	Разбор примеров, решение задач.
2.12	Стандартная библиотека.	Разбор примеров, решение задач.

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

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)				
		Лекции	Практические	Лабораторные	Самостоятельная работа	Всего
1	Введение. Ввод и вывод.	2	0	7	12	21
2	Типы и объявления.	2	0	7	12	21
3	Указатели, массивы и структуры.	4	0	11	12	27
4	Выражения и инструкции.	2	0	7	12	21
5	Функции.	4	0	11	12	27
6	Исходные файлы. Компиляция.	2	0	7	12	21
7	Классы.	4	0	5	12	21
8	Перегрузка операций.	4	0	5	12	21
9	Наследование классов. Виртуальные функции.	4	0	5	12	21
10	Шаблоны.	4	0	3	19	26
11	Пространства имен. Обработка исключений.	4	0	3	12	19
12	Стандартная библиотека.	14	0	25	42	81
	Итого:	50	0	94	180	324

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

Работа с конспектами лекций, выполнение практических заданий для самостоятельной работы, выполнение лабораторных работ, использование рекомендованной литературы и методических материалов. Выполнение контрольных работ.

#### 15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	Страуструп Б. Язык программирования С++. Специальное издание / Бьерн Страуструп; пер. с англ. под ред. Н.Н. Мартынова. — Москва. Бинوم, 2015. — 1135 с. : ил.
2	Подбельский, В.В. Курс программирования на языке Си [Электронный ресурс]: учебник / В.В. Подбельский, С.С. Фомин. — Электрон, дан. — М.: ДМК Пресс, 2012. — 384 с. — Режим доступа: <a href="http://e.lanbook.com/books/element.php?pl1_id=4148">http://e.lanbook.com/books/element.php?pl1_id=4148</a>
3	Страуструп, Б. Дизайн и эволюция С++ [Электронный ресурс] / Б. Страуструп; Пер. с англ. - М.: ДМК Пресс, 2007. - 448 с.: ил. <a href="http://znanium.com/catalog.php?bookinfo=409529">http://znanium.com/catalog.php?bookinfo=409529</a>

б) дополнительная литература:

№ п/п	Источник
4	Саттер Г. Решение сложных задач на С+: 87 головоломных примеров с решениями / Герб Саттер; Пер. с англ. и ред. И.В.Красикова; Предисл. Э.Кенига. — М. Вильямс, 2002. — 395 с. ил.
5	Мейерс С. Эффективное использование STL / Скотт Мейерс; Пер. с англ. Е. Матвеева. — СПб. Питер, 2002. — 224 с. ил.
6	Александреску А. Современное проектирование на С++: Обобщенное программирование и прикладные шаблоны проектирования / Андрей Александреску; Пер. с англ. и ред. Д.А. Ключина. — Изд. испр. — М. Вильямс, 2004. — 335 с. ил.
7	Хохлов, Д.Г. Методы программирования на языке С: практикум. Ч.1 [Электронный ресурс]: учебное пособие. — Электрон. дан. — М. Бинум. Лаборатория знаний, 2014. — 336 с. — Режим доступа: <a href="http://e.lanbook.com/books/element.php?pl1_id=50555">http://e.lanbook.com/books/element.php?pl1_id=50555</a>
8	Хохлов, Д.Г. Методы программирования на языке С: практикум. Ч.2 [Электронный ресурс]: — Электрон. дан. — М. Бинум. Лаборатория знаний, 2014. — 377 с. — Режим доступа: <a href="http://e.lanbook.com/books/element.php?pl1_id=50556">http://e.lanbook.com/books/element.php?pl1_id=50556</a>
9	Шилдт Г. Теория и практика С+ / Герберт Шилдт; Пер. с англ. и науч. ред. О. Кокорева. — СПб. и др. : ВHV, 2000. — 412с.
10	Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре; [пер. с англ. А. Кузнецова, М. Назарова, В. Шрага]. — 4-е изд. — Санкт-Петербург: Питер, 2015. — 923 с.: ил. табл.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
11	Электронный каталог Научной библиотеки Воронежского государственного университета. — ( <a href="http://www.ru/lib.vsu/ru">http://www.ru/lib.vsu/ru</a> )

#### 16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник

**17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационносправочные системы (при необходимости)**

ОС Windows, Microsoft Visual Studio

**18. Материально-техническое обеспечение дисциплины:**

Аудитория, доска, ЭВМ

**19. Фонд оценочных средств:**

**19.1. Перечень компетенций с указанием этапов формирования и планируемых результатов обучения**

Код и содержание компетенции (или ее части)	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков)	Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование)	ФОС* (средства оценивания)
ОПК-3 готовность анализировать проблемы и направления развития технологий программирования	Знать: основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.	Разделы 1.1 – 1.12	Комплект КИМ №1-2
	Уметь: проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.	Разделы 2.1-2.12	Контрольные работы №1-3
	Владеть: навыками практического программирования конкретных задач с использованием языка программирования C/C++.	— « —	Комплект КИМ №1-2

ОПК-7 способность использовать знания основных концептуальных положений функционального, логического, объектноориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	Знать: основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.	Разделы 1.1 – 1.12	Комплект КИМ №1-2
	Уметь: проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный	Разделы 2.1-2.12	Контрольные работы №1-3
	контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.		
	Владеть: навыками практического программирования конкретных задач с использованием языка программирования C/C++.	— « —	Комплект КИМ №1-2
ОПК-8 способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание ПО	Знать: основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектноориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.	Разделы 1.1 – 1.12	Комплект КИМ №1-2
	Уметь: проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.	Разделы 2.1-2.12	Контрольные работы №1-3
	Владеть: навыками практического программирования конкретных задач с использованием языка программирования C/C++.	— « —	Комплект КИМ №1-2



ОПК-11 готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	Знать: основные управляющие конструкции языка C/C++, типы данных и операторы языка C++, методы проектирования приложений на языке C/C++, принципы объектно-ориентированного и обобщенного программирования на языке C/C++, типы контейнеров и алгоритмы стандартной библиотеки.	Разделы 1.1 – 1.12	Комплект КИМ №1-2
	Уметь: проектировать и реализовать приложение на языке C/C++, в том числе с использованием методов ООП, выбрать оптимальный контейнер данных и реализовать приложение с использованием алгоритмов стандартной библиотеки.	Разделы 2.1-2.12	Контрольные работы №1-3
	Владеть: навыками практического программирования конкретных задач с использованием языка программирования C/C++.	— « —	Комплект КИМ №1-2
<b>Промежуточная аттестация</b>			Комплект КИМ №1-2

## 19.2 Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

### Пример:

Для оценивания результатов обучения на экзамене/зачете используются следующие показатели:

- 1) владение навыками реализации программ с использованием языка C++;
- 2) знание методологии объектно-ориентированного программирования и применение ее на практике;
- 3) знание и умение использовать при решении задач базовые типы данных языка C++;
- 4) умение выбирать и применять при реализации оптимальный контейнер данных;
- 5) умение разрабатывать оптимальный алгоритм решения задачи и выполнять его реализацию;
- 4) знание теоретического материала.

Для оценивания результатов обучения на экзамене (зачете с оценкой) используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
Отличное знание теоретического материала, правильное и эффективное решение задачи, правильный ответ на тест. Должны быть выполнены ВСЕ лабораторные работы.	<i>Повышенный уровень</i>	<i>Отлично</i>
Хорошее знание теоретического материала, в целом правильное решение задачи. НО: допускает незначительные ошибки в решении задачи. Неправильный ответ на тест. ИЛИ: выполнены все показатели повышенного уровня, но не зачтена одна лабораторная работа, но студент продемонстрировал умение решать задачи по этой теме (это задача в КИМе)	<i>Базовый уровень</i>	<i>Хорошо</i>
Решение задачи не доведено до конца или недостаточное знание теоретического материала. Неоптимальное решение задачи и недостаточное владение теоретическим материалом. Неправильный тест.	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
ИЛИ: выполнены все показатели базового уровня, но не зачтено более одной лабораторной работы.		
Задача не решена или серьезные пробелы в знании теоретического материала (с незнанием могут быть связаны и грубые ошибки в ответе на тестовые вопросы).	–	<i>Неудовлетворительно</i>

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

#### 19.3.1 Перечень вопросов к экзамену:

##### Экзамен №1

1. Логический тип данных.
2. Арифметические типы данных.
3. Символьные типы данных.
4. Размеры фундаментальных типов данных.
5. Перечисления.
6. Объявления и имена.
7. Инициализация в объявлении.
8. Понятие области видимости.
9. Ключевое слово typedef.
10. Тип данных void.

11. Указатели, нулевой указатель.
12. Массивы. Инициализация массивов.
13. Строковые литералы.
14. Константы.
15. Ссылки.
16. Структуры.
17. Выражения. Последовательность вычисления выражений.
18. Приоритет операций.
19. Арифметические операции.
20. Логические операции.
21. Побитовые операции.
22. Операции инкремента и декремента.
23. Операции работы с памятью new и delete.
24. Операции приведения типов.
25. Условные операторы if, if else, switch.
26. Операторы цикла for, while, do while.
27. Оператор goto.
28. Комментарии и отступы.
29. Понятие функции. Объявление функции.
30. Передача аргументов в функцию.
31. Возврат значения из функции, оператор return.
32. Перегрузка имен функций.
33. Правила разрешения перегрузки в случае одного аргумента.
34. Правила разрешения перегрузки в случае нескольких аргументов.
35. Аргументы функции по умолчанию.
36. Функции с переменным числом аргументов.
37. Указатели на функции.
38. Общая схема программы. Заголовочные файлы.
39. Компиляция. Внешние заголовочные файлы.
40. Компоновка. Правило одного определения.
41. Классы и объекты. Переменные и функции члены классов.
42. Управление режимом доступа к членам классов.
43. Конструкторы класса.
44. Статические члены класса.
45. Копирование объектов класса.
46. Константные функции-члены.
47. Структуры и классы.
48. Определение функций членов класса.

49. Функции-операции класса.
50. Перегрузка унарных операций класса.
51. Перегрузка бинарных операций класса.
52. Дружественные функции класса.
53. Операция индексирования operator[].
54. Операция функционального вызова operator().
55. Операции инкремента и декремента.
56. Наследование классов. Производные классы.
57. Конструирование и уничтожение объектов при наследовании классов.
58. Контроль доступа при наследовании классов.
59. Полиморфизм. Виртуальные функции.
60. Абстрактные классы.
61. Множественное наследование.
62. Виртуальное наследование.
63. Статическое и динамическое приведение типов.
64. Операция typeid.
65. Определение шаблона класса.
66. Конкретизация шаблона.
67. Параметры шаблонов.
68. Определение функциональных шаблонов.
69. Аргументы функциональных шаблонов.
70. Перегрузка функциональных шаблонов.
71. Параметры шаблонов по умолчанию.
72. Специализация шаблонов.
73. Объявление пространств имен.
74. Использование пространств имен.
75. Генерация исключений. Ключевое слово throw.
76. Обработка исключений. Ключевые слова try и catch.
77. Обработка нескольких исключений.

## **Экзамен №2**

1. Стандартная библиотека C++. Назначение, структура и основные принципы организации.
2. Контейнер vector. Назначение и основные принципы устройства контейнера.
3. Контейнер vector. Типы данных, итераторы, доступ к элементам, конструирование. Размер и емкость.
4. Контейнер vector. Стековые и списочные операции.
5. Специализация vector<bool>.
6. Контейнеры стандартной библиотеки. Классификация и перечень.
7. Контейнеры стандартной библиотеки. Требования к элементам контейнеров.
8. Контейнер list. Назначение, принципы устройства, основные операции.

9. Контейнер deque. Назначение, принципы устройства, основные операции.
10. Адаптеры стандартных контейнеров. Стек.
11. Адаптеры стандартных контейнеров. Очередь.
12. Адаптеры стандартных контейнеров. Очередь с приоритетом.
13. Ассоциативный контейнер map. Назначение, принципы устройства, основные операции.
14. Ассоциативный контейнер set. Назначение, принципы устройства, основные операции.
15. Ассоциативные контейнеры multimap и multiset. Назначение, принципы устройства и основные отличия от контейнеров map и set.
16. Понятие итератора. Основные принципы.
17. Категории итераторов.
18. Обратные итераторы.
19. Поточковые итераторы (ввод и вывод).
20. Понятие аллокатора. Основные принципы.
21. Алгоритмы стандартной библиотеки. Основные принципы и классификация.
22. Функциональные объекты (функторы). Предикаты.
23. Немодифицирующие алгоритмы (for\_each, count, equal, mismatch).
24. Немодифицирующие поисковые алгоритмы.
25. Модифицирующие алгоритмы (копирующие алгоритмы, transform, unique).
26. Модифицирующие алгоритмы (замена и удаление элементов).
27. Модифицирующие алгоритмы (fill, generate, reverse, rotate).
28. Алгоритмы сортировки. Бинарный поиск.
29. Алгоритмы слияния и разбиения (merge, partition).
30. Алгоритмы, реализующие операции над множествами (includes, set\_union, set\_intersection, set\_difference, set\_symmetric\_difference).

### 19.3.2 Перечень практических заданий

Иллюстрируется на примере КИМ №1 и КИМ №2

### 19.3.4 Тестовые задания

Иллюстрируется на примере КИМ №1 и КИМ №2

#### ПРИМЕРЫ КИМ №1

<b>Вопрос</b>	Логический тип данных. Арифметические типы данных.
<b>Вопрос</b>	Классы и объекты. Переменные и функции члены классов. Управление режимом доступа к членам классов.
<b>Задача</b>	Реализовать структуру для работы с точками на плоскости в декартовой системе координат. Реализовать функции перемещение точки по оси X, перемещение точки по оси Y, определение расстояния между двумя точками, сравнение двух точек на равенство, расстояние до начала координат, перевод в полярные координаты.

<b>Тест</b>	<p>Каков размер массива str в следующем примере:</p> <pre>char str[] = "a short string";</pre> <p>Какова длина строки "a short string11"?</p>
-------------	---

### КИМ №2

<b>Вопрос</b>	Тип данных void. Символьные типы данных. Размеры фундаментальных типов данных.
<b>Вопрос</b>	Конструкторы и деструктор класса.
<b>Задача</b>	Реализовать структуру для работы с комплексными числами. Реализовать функции сложения, вычитания, умножения, деления комплексных чисел, вычисления сопряженного числа, а также сравнения (равно/не равно) двух комплексных чисел.
<b>Тест</b>	<p>Расставьте скобки в следующих выражениях:</p> <p>a) *p++</p> <p>b) *--p</p> <p>c) ++a--</p> <p>d) (int*) p-&gt;m</p> <p>e) *p.m</p> <p>f) *a[i]</p>

### 19.3.4 Перечень заданий для контрольных работ

Иллюстрируется на примерах заданий для контрольных работ 1-3

**Примеры заданий для контрольной работы № 1 Необходимо выбрать все верные варианты ответов (их может быть несколько на один вопрос):**

#### Вариант 1

1. Разделение программы на функции:
  - a. является ключевым методом объектно-ориентированного программирования;
  - b. упрощает представление программы;
  - c. сокращает размер программного кода;
  - d. ускоряет процесс выполнения программы.
2. Операция отношения:
  - a. присваивает значение одного операнда другому операнду;
  - b. имеет своим результатом булево значение;
  - c. сравнивает значения двух операндов;
  - d. создает логическую комбинацию двух операндов.
3. Структура объединяет:
  - a. данные одного типа;
  - b. логически связанные данные;
  - c. целые именованные значения;
  - d. переменные.

## Вариант 2

1. Выражение:
  - a. всегда приводит к вычислению значения;
  - b. является способом высказывания программы;
  - c. всегда происходит вне функции;
  - d. является частью оператора.
2. Переменная описанная внутри блока видима:
  - a. от точки своего объявления до конца программы;
  - b. от точки своего объявления до конца функции;
  - c. от точки своего объявления до конца блока;
  - d. внутри функции.
3. При обращении к полю структуры левым операндом операции (.) является:
  - a. поле структуры;
  - b. имя структуры;
  - c. структурная переменная;
  - d. ключевое слово struct. **Вариант 3**

1. В 32-разрядной системе тип int имеет размер:
  - a. 8 байт;
  - b. 1 байт;
  - c. 2 байта;
  - d. 4 байта.
2. Операции && и ||:
  - a. сравнивают два численных значения;
  - b. комбинируют два численных значения;
  - c. сравнивают два булевых значения;
  - d. комбинируют два булевых значения.
3. Перечисление объединяет:
  - a. данные различных типов;
  - b. логически связанные переменные;
  - c. именованные целые значения;
  - d. константные значения. **Вариант 4**

1. Литерал 3.1415 имеет тип:
  - a. int;
  - b. float;
  - c. double;
  - d. тип зависит от реализации компилятора.
2. Оператор break производит выход:
  - a. только из цикла наибольшей глубины вложенности;
  - b. только из ветвления switch наибольшей глубины вложенности;
  - c. из всех вложенных циклов и ветвлений;
  - d. из циклов и ветвлений наибольшей глубины вложенности.
3. Структурные переменные:
  - a. можно присваивать друг другу;
  - b. можно передавать в качестве аргументов функций;
  - c. можно сравнивать с использованием логических операций;
  - d. можно выводить на экран стандартными средствами.

### Критерии оценки:

- оценка «зачтено» выставляется студенту, если он правильно ответил на два задания;
- оценка «не зачтено» выставляется студенту, если он неверно ответил на два или более задания.

## Примеры заданий для контрольной работы № 2

Необходимо выполнить задание и дать ответ на вопрос.

### Вариант 1

1. Напишите функции foo, выводящую на экран фразу «Hello world».
2. Как называется первая строка в объявлении функции?

### Вариант 2

1. Напишите прототипы двух перегруженных функций с именем bar(). Обе функции имеют возвращаемое значение типа int. Первая функция имеет единственный аргумент типа char, а вторая – два аргумента типа char. Если это невозможно, объясните причину.
2. Для чего предназначена возможность указывать в прототипе функции имена аргументов?

### Вариант 3

1. Напишите прототип функции foo, возвращающей значение типа char и принимающей два аргумента. Первый аргумент имеет тип char, а второй тип float и значение по умолчанию 3.14159.
2. Какие функции имеют доступ к локальной переменной?

### Вариант 4

1. Напишите прототип встроенной функции с именем foobar, имеющей один аргумент типа float и возвращающей значение типа float.
2. Какова причина использования ссылочного механизма передачи аргументов в функцию?

#### Критерии оценки:

- оценка «зачтено» выставляется студенту, если он правильно выполнил задание и ответил на вопрос;
- оценка «не зачтено» выставляется студенту, если он неверно выполнил задание или неправильно ответил на вопрос.

## Примеры заданий для контрольной работы № 3

Необходимо выполнить задание.

### Вариант 1

Создайте класс Int, имитирующий стандартный тип int. Единственное поле этого класса должно иметь тип int. Класс должен иметь конструктор, позволяющий инициализировать поле класса значением. Создайте методы, которые будут выводить значение поля на экран и выполнять базовые арифметические операции над значением: сложение, вычитание, умножение, деление.

### Вариант 2

Создайте класс Time, содержащий три поля типа int, предназначенные для хранения часов, минут и секунд. Один из конструкторов класса должен инициализировать поле нулевыми значениями, а другой конструктор – заданным набором значений. Создайте метод класса, который будет выводить значение полей на экран в формате HH:MM:SS и метод, складывающий значение двух объектов типа Time, передаваемых в качестве аргументов.



### **Вариант 3**

Создайте класс Date, содержащий три поля типа int, предназначенные для хранения года, месяца и дня. Один из конструкторов класса должен инициализировать поле нулевыми значениями, а другой конструктор – заданным набором значений. Создайте метод getDate, который будет возвращать строковое представление даты в формате DD/MM/YY.

### **Вариант 4**

Создайте класс Fraction, содержащий два поля типа int, предназначенные для числителя и знаменателя дроби. Создайте конструктор для инициализации полей значениями. Создайте методы, которые будут выводить значение дроби на экран и выполнять базовые арифметические операции над дробями: сложение, вычитание, умножение

#### **Критерии оценки:**

- оценка «отлично» выставляется студенту, если все условия задания выполнены и ошибки отсутствуют;
- оценка «хорошо» выставляется студенту, если все условия задания выполнены с незначительными ошибками;
- оценка «удовлетворительно» выставляется студенту, если не выполнена часть условий, а выполненные условия не содержат существенных ошибок;
- оценка «неудовлетворительно» выставляется студенту, если не выполняется ни один из вышеуказанных критериев оценки.

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

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 форме контрольных работ.

Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя практические задания, позволяющие оценить степень сформированности умений и навыков.

При оценивании используются количественные шкалы оценок. Критерии оценивания приведены выше.