

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.

02.04.2024 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

Б1.О.28 Языки и системы программирования

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 (степень) выпускника: бакалавр

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины:
ПОиАИС

6. Составители программы: Ефремов М. С.

7. Рекомендована: НМС факультета ПММ протокол №5 от 22.03.2024

8. Учебный год: 2024/2025

Семестр(ы): 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 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. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):

Компетенция		Планируемые результаты обучения
Код	Название	
ОПК-3	Способность использовать и применять углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий	Знать основные управляющие конструкции языка C/C++, типы данных и операторы языка C/C++, методы проектирования приложений на языке C/C++, принципы объектно-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++.
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	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. 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++.
ОПК-8	Способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения	Знать теоретические основы проектирования и разработки программ с использованием C++ Уметь проектировать и разрабатывать программы с использованием C++ Владеть средствами разработки и отладки программ C++.

ОПК-11	Готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях	Знать теоретические основы проектирования программ, оценки качества и эффективности программ, преимущества и недостатки C++, критерии выбора необходимых для решения задачи технологий. Уметь выбирать подходящие средства разработки программ, оценивать преимущества и недостатки использования языка программирования C++ для решения поставленных задач, оценивать качество программного кода, используемую память и производительность разрабатываемого приложения. Владеть навыками практического программирования с использованием языка программирования C++ и сопутствующих программных средств.
--------	---	--

12. Объем дисциплины в зачетных единицах/часах в соответствии с учебным планом — 7/252.

Форма промежуточной аттестации: экзамен

13. Виды учебной работы

Вид учебной работы	Трудоемкость (часы)		
	Всего	По семестрам	
		3	4
Аудиторные занятия	96	48	48
в том числе: лекции	64	32	32
лабораторные	32	16	16
практические			
Самостоятельная работа	84	24	60
Итого	252	108	144
Контроль:	72	36	36
Итого:	252	108	144
Форма промежуточной аттестации		Экзамен	экзамен

13.1. Содержание дисциплины

п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1.1	Введение. Ввод и вывод.	Обзор возможностей языка C++. Базовые средства ввода-вывода. 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, выведение 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.
2. Лабораторные работы		
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

	Виды занятий (часов)
--	----------------------

№ п/п	Наименование темы (раздела) дисциплины	Лекции	Практически е	Лабораторные	Самостоятельная работа	Всего
1	Введение. Ввод и вывод.	2	0	2	7	11
2	Типы и объявления.	2	0	2	7	11
3	Указатели, массивы и структуры.	4	0	2	7	13
4	Выражения и инструкции.	2	0	2	7	11
5	Функции.	4	0	2	7	13
6	Исходные файлы. Компиляция.	2	0	2	7	11
7	Классы.	4	0	2	7	13
8	Перегрузка операций.	4	0	2	7	13
9	Наследование классов. Виртуальные функции.	4	0	2	7	13
10	Шаблоны.	14	0	6	7	27
11	Пространства имен. Обработка исключений.	8	0	2	7	6
12	Стандартная библиотека.	14	0	6	7	27
	Итого:	64	0	32	84	252

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. Бином, 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: http://e.lanbook.com/books/element.php?pl1_id=4148
3	Страуструп, Б. Дизайн и эволюция С++ [Электронный ресурс] / Б. Страуструп; Пер. с англ. - М.: ДМК Пресс, 2007. - 448 с.: ил. http://znaniy.com/catalog.php?bookinfo=409529

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. Бином. Лаборатория знаний, 2014. — 336 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=50555
8	Хохлов, Д.Г. Методы программирования на языке С: практикум. Ч.2 [Электронный ресурс]: — Электрон. дан. — М. Бином. Лаборатория знаний, 2014. — 377 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=50556
9	Шилдт Г. Теория и практика С+ / Герберт Шилдт; Пер. с англ. и науч. ред. О. Кокорева. — СПб. и др. : BHV, 2000. — 412с.
10	Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре; [пер. с англ. А. Кузнецова, М. Назарова, В. Шпара]. — 4-е изд. — Санкт-Петербург: Питер, 2015. — 923 с.: ил. табл.

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет) *:

№ п/п	Ресурс
-------	--------

11	Электронный каталог Научной библиотеки Воронежского государственного университета. – (http://www.ru/lib.vsu/ru)
----	---

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Ресурс
11	Электронный каталог Научной библиотеки Воронежского государственного университета. – (http://www.ru/lib.vsu/ru)

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение и информационно-справочные 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 Способность использовать и применять углубленные теоретические и практические знания в области фундаментальной информатики и информационных технологий	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	Знать: основные управляющие конструкции языка C/C++, типы данных и операторы языка C++,	Разделы 1.1 – 1.12	Комплект КИМ №1-2

концептуальных положений функционального, логического, объектно-ориентированного и визуального направлений программирования, методов, способов и средств разработки программ в рамках этих направлений	методы проектирования приложений на языке C/C++, принципы объектно-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.	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-8 способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание ПО	Знать: теоретические основы проектирования и разработки программ с использованием C++.	Разделы 1.1 – 1.12	Комплект КИМ №1-2
	Уметь: проектировать и разрабатывать программы с использованием C++.	Разделы 2.1-2.12	Контрольные работы №1-3
	Владеть: средствами разработки и отладки программ 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: теоретические основы проектирования программ, оценки качества и эффективности программ, преимущества и недостатки C++, критерии выбора необходимых для решения задачи технологий.	Разделы 1.1 – 1.12	Комплект КИМ №1-2
	Уметь: выбирать подходящие средства разработки программ, оценивать преимущества и недостатки использования языка программирования C++ для решения поставленных задач, оценивать качество программного кода, используемую память и производительность разрабатываемого приложения.	Разделы 2.1-2.12	Контрольные работы №1-3
	Владеть: навыками практического программирования с использованием языка программирования C++ и сопутствующих программных средств.	— « —	Комплект КИМ №1-2
Промежуточная аттестация			Комплект КИМ №1-2

--	--

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

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;

Для оценивания результатов обучения на 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. ИЛИ: выполнены все показатели базового уровня, но не зачтено более одной лабораторной работы.	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
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. Поточные 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.

19.3.3 Тестовые задания

Иллюстрируется на примере КИМ №1 и КИМ №2.

КИМ №1

Вопрос	Логический тип данных. Арифметические типы данных.
--------	--

Вопрос	Классы и объекты. Переменные и функции члены классов. Управление режимом доступа к членам классов.
Задача	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.
Тест	Каков размер массива str в следующем примере: char str[] = "a short string"; Какова длина строки "a short string11"?

КИМ №2

Вопрос	Тип данных void. Символьные типы данных. Размеры фундаментальных типов данных.
Вопрос	Конструкторы и деструктор класса.
Задача	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.
Тест	Расставьте скобки в следующих выражениях: a) *p++ b) *--p c) ++a-- d) (int*) p->m e) *p.m f) *a[i]

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.

19.5 Темы курсовых работ

1. Модульное программирование. Задача о векторах
3. Модульное программирование. Задача о комплексных числах
4. Модульное программирование. Задача о «больших» множествах
5. Модульное программирование. Задача о длинных строках
6. Модульное программирование. Задача о датах
7. Модульное программирование. Задача о длинных числах
8. Модульное программирование. Задача о полиномах
9. Модульное программирование. Список с заглавным звеном
10. Модульное программирование. Список без заглавного звена
11. Стеки. Цепочное представление
12. Стеки. Сплошное представление

13. Очереди. Цепочное представление
14. Очереди. Сплошное представление
15. Бинарные деревья. Случайные деревья (цепочное и сплошное представление)
16. Бинарные деревья. Бинарные деревья поиска (цепочное и сплошное представление)
17. Сбалансированные бинарные деревья поиска (АВЛ-деревья).
18. Trie-деревья. Задача обо всех словах, удовлетворяющих условию отбора
19. Trie-деревья. Задача о количестве слов, удовлетворяющих условию отбора
20. Trie-деревья. Задача об удалении слов, удовлетворяющих условию отбора
21. B-деревья
22. B+-деревья
23. Алгоритмы с возвратом. Поиск всех возможных решений
24. Алгоритмы с возвратом. Поиск одного из возможных решений
25. Алгоритмы с возвратом. Поиск лучшего решения
26. Хеширование. Внутренние цепочки
27. Хеширование. Внешние цепочки
28. Хеширование. Методы опробывания (линейное и квадратичное)
29. Хеширование. Двойное хеширование
30. Оценка качества хэш-функции.
31. Внутренние сортировки. Сортировки методом подсчета.
32. Внутренние сортировки. Простые обменные сортировки.
33. Внутренние сортировки. Простейшие сортировки вставками (простые и бинарные вставки)
34. Внутренние сортировки. Вставки в список (простые и двухпутевые: с помощью массива, с помощью списка).
35. Внутренние сортировки. Вставки в список с вычислением адреса.
36. Внутренние сортировки. Быстрая сортировка (сортировка Хоара) (два метода).
37. Внутренние сортировки. Обменная поразрядная сортировка.
38. Внутренние сортировки. Параллельная сортировка Бетчера.
39. Внутренние сортировки. Сортировка с убывающим шагом (сортировка Шелла).
40. Внутренние сортировки. Пирамидальная сортировка.
41. Внутренние сортировки. Простое слияние.
42. Внутренние сортировки. Естественное слияние.
43. Внутренние сортировки. Турнирная сортировка.
44. Внешние сортировки. Двухпутевое двухфазное простое слияние (без внутренней сортировки или с ней).
45. Внешние сортировки. Многопутевое двухфазное простое слияние (без внутренней сортировки или с ней).
46. Внешние сортировки. Двухпутевое однофазное простое слияние (без внутренней сортировки или с ней).
47. Внешние сортировки. Многопутевое однофазное простое слияние (без внутренней сортировки или с ней).
48. Внешние сортировки. Двухпутевое двухфазное естественное слияние (сбалансированное, несбалансированное).
49. Внешние сортировки. Многопутевое двухфазное естественное слияние (сбалансированное, несбалансированное).
50. Внешние сортировки. Двухпутевое однофазное естественное слияние (сбалансированное, несбалансированное).
51. Внешние сортировки. Многопутевое однофазное естественное слияние (сбалансированное, несбалансированное).
52. Внешние сортировки. Многофазная сортировка.
53. Внешние сортировки. Каскадная сортировка.

Требования к реализации курсовых проектов

1. Грамотное изложение теоретического материала.
2. Грамотное применение современных технологий и средств проектирования.
3. Правильное оформление работы:
 - соответствие структуре работы;
 - соблюдение правил оформления работы;
 - грамотность.

Курсовая работа подразумевает обязательную публичную защиту, ответы на вопросы комиссии.

Шкала оценок	
Отлично	Грамотно и четко сформулирована постановка задачи, продемонстрирован высокий уровень готовности использовать навыки выбора, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях, продемонстрирован высокий уровень готовности к использованию основных моделей информационных технологий и способов их применения для решения задач в предметных областях, выявлена ярко выраженная способность к самоорганизации и самообразованию, четко и качественно изложен материал работы, четко и квалифицированно даны ответы на все дополнительные вопросы, отзыв носит положительный характер
Хорошо	Корректно сформулирована постановка задачи, продемонстрирована готовность использовать навыки выбора, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях, продемонстрирована готовность к использованию основных моделей информационных технологий и способов их применения для решения задач в предметных областях, выявлена способность к самоорганизации и самообразованию, четко и качественно изложен материал работы, не на все дополнительные вопросы даны исчерпывающие ответы, имеются претензии к объему выполненной работы, отзыв носит положительный характер
Удовлетворительно	Компетентность в предметной области продемонстрирована недостаточно, постановка задачи сформулирована расплывчато, недостаточно четко продемонстрирована готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях, выявлены незначительные пробелы в готовности к использованию основных моделей информационных технологий и способов их применения для решения задач в предметных областях, выявлен невысокий уровень способностей к самоорганизации и самообразованию, изложение материала работы содержит нечеткие формулировки и является непоследовательным, ответы на

	дополнительные вопросы неполные или содержат неточности и ошибочные утверждения, дан положительный отзыв
Неудовлетворительно	Низкий уровень компетентности в предметной области, постановка задачи сформулирована нечетко и с погрешностями, низкий уровень теоретической и практической подготовки, недостаточное владение или неиспользование современных информационных технологий, изложение материала работы содержит нечеткие формулировки и ошибочные утверждения, даны неверные ответы на дополнительные вопросы

Замечание: неправильное оформление работы может повлечь за собой существенное снижение оценки (на 1-2 балла), в зависимости от степени невыполнения требований.