

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

УТВЕРЖДАЮ
заведующий кафедрой
кибербезопасности
информационных систем
С.Л. Кенин



17.03.2025

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.06 Разработка приложений на C++

1. Код и наименование направления подготовки/специальности:

10.05.01 Компьютерная безопасность

2. Специализация:

Математические методы защиты информации

3. Квалификация (степень) выпускника: специалист по защите информации

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины:

кибербезопасности информационных систем

6. Составители программы:

Сафронов Виталий Владимирович, к.т.н., доцент кафедры кибербезопасности информационных систем

7. Рекомендована:

НМС факультета ПММ, протокол № 6 от 17.03.2025г.

8. Учебный год: 2026/2027

Семестр(ы): 3

9. Цели и задачи учебной дисциплины

Целями освоения учебной дисциплины являются:

приобретение базовых знаний и навыков по алгоритмизации, разработке, отладке и тестированию программ на языке C++, проектированию и разработке приложений с применением объектно-ориентированного подхода.

Задачи учебной дисциплины:

- изучение технологии программирования на языке C++;
- раскрытие принципов объектно-ориентированного подхода при проектировании и разработке приложений;
- овладение средствами объектно-ориентированного и обобщенного программирования языка C++, средствами стандартной библиотеки STL;
- изучение методов отладки и тестирования программ на C++.

10. Место учебной дисциплины в структуре ОПОП: дисциплина относится к вариативной части блока Б1 дисциплин учебного плана.

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)	Индикаторы(ы)	Планируемые результаты обучения
ПК-1	Способен проводить анализ требований и выполнять работы по проектированию программного обеспечения с применением математических методов защиты	ПК-1.1	Применяет различные методы разработки программного обеспечения и технологии программирования	Знает: современные методы разработки программного обеспечения и технологии программирования; Умеет: применять современные методы разработки программного обеспечения и технологии программирования; использовать принципы комплексной разработки правил, процедур, приемов и методов, при создании средств защиты информации, в том числе с использованием современных методов и средств разработки программного обеспечения Владеет современными методами разработки программного обеспечения и технологии программирования;
		ПК-1.2	Использует современные математические методы и алгоритмы функционирования для компонентов программных средств	
		ПК-1.3	Применяет технологии обработки данных, анализирует возможности их использования при разработке программного обеспечения профессиональной деятельности	
ПК-2	Способен проводить исследования на всех этапах жизненного цикла средств защиты информации в профессиональной деятельности	ПК-2.3	Использует типовое и специализированное программное обеспечение, проводит компьютерное исследование, формирует описание результатов и формулирует выводы	
ПК-3	Способен осуществлять разработку, анализ и	ПК-3.4	Разрабатывает программные алгоритмы, реализующие	

	обосновывать эффективность применяемых математических методов защиты информации, возникающих при работе программных и программно-аппаратных средств защиты информации при решении профессиональных, исследовательских и прикладных задач		современные математические методы защиты информации.	
--	--	--	--	--

12. Объем дисциплины в зачетных единицах/час - 3/108.

Форма промежуточной аттестации – зачет с оценкой.

13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоёмкость (часы)				
	Всего	В том числе в интерактивной форме	По семестрам		
			3		
Аудиторные занятия	28		28		
в том числе: лекции	14		14		
Практические	0		0		
Лабораторные	14		14		
Самостоятельная работа	44		44		
Контроль	36		36		
Итого:	108		108		
Форма промежуточной аттестации	Экзамен		Экзамен		

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Понятие алгоритма, алгоритмического мышления и информации. Понятие алгоритмической ошибки.	Понятие алгоритма, запись в виде схемы алгоритма. Алгоритмические ошибки и причины их возникновения. Проблема переполнения.	https://edu.vsu.ru/course/
1.2	Основные типы данных, операторы языка С. Основные функции стандартной библиотеки. Файлы. Понятие инварианта цикла.	Концепции языка С. Целые и вещественные типы данных. Представление целых чисел в С, особенности операций над вещественными числами. Оператор sizeof. Функции ввода-вывода, работы с файлами, математические функции стандартной библиотеки. Инварианты в циклических алгоритмах.	

1.3	Специфика языка С. Массивы и указатели, процедуры и функции. Строки в С и функции работы со строками.	Связь массивов и указателей в С, операции над указателями. Передача параметров в функции и процедуры, локальные и глобальные переменные. Понятие строки символов и отличие от символьного массива. Функции strlen, strcpy, strcat. strcmp и их сложность	
1.4	Основные алгоритмы и их сложность. Алгоритмы поиска, сортировки, поиска подстроки в строке.	Линейный и бинарный поиск, поиск с барьером. Сортировки вставками, поиском максимума, пузырьковая сортировка. Улучшения простых сортировок. Быстрая сортировка. Линейный поиск подстроки в строке, алгоритм Бойера-Мура. Понятие сложности алгоритмов, O-нотация и вывод сложности для алгоритмов поиска и сортировки.	
1.5	Понятие рекурсии и рекурсивных алгоритмов. Алгоритмы перебора с возвратом, алгоритмы оптимального поиска.	Определение рекурсии. Виды и характеристики рекурсии. Простые рекурсивные алгоритмы - перестановки массива, задача о Ханойских башнях. Алгоритмы перебора с возвратом с использованием рекурсии. Дерево перебора, способы сокращения перебора. Алгоритмы оптимального поиска на примере задачи о коммивояжере. Метод ветвей и границ; модификация Литтла. Понятие о P- и NP-сложных задачах.	
1.6	Псевдослучайные числа. Алгоритмы генерации ПСЧ, проверки качества генераторов. Алгоритмы, использующие случайные числа.	Алгоритм Кнута. Линейный конгруэнтный генератор и его свойства. Правильный выбор параметров ЛКГ. Тесты на случайность. Алгоритм случайной перестановки и случайного выбора.	
2. Лабораторные работы			
2.1	Лабораторная работа 1.	Рекурсивный алгоритм «ханойские башни».	https://edu.vsu.ru/course/
2.2	Лабораторная работа 2	Метод барьера на шахматной доске. Тестирование модульной программы с помощью автоматизированной системы.	
2.3	Лабораторная работа 3	Реализация линейного конгруэнтного генератора.	
2.4	Лабораторная работа 4	Реализация алгоритма оптимального поиска с помощью метода ветвей и границ (задача о коммивояжере и т.п.)	
2.5	Лабораторная работа 5	Практическое выяснение параметров быстрой сортировки.	
2.6	Лабораторная работа 6	Рекурсивная функция вычисления символа Якоби	

13.2. Темы (разделы) дисциплины и виды занятий

№ п/п	Наименование раздела дисциплины	Виды занятий (часов)					
		Лекции	Практ.	Лаб. раб.	Самостоятельная работа	Контроль	Всего
1.1	Понятие алгоритма, алгоритмического мышления и информации. Понятие алгоритмической ошибки.	2		2	6	6	16
1.2	Основные типы данных, операторы языка С. Основные функции стандартной библиотеки. Файлы. Понятие инварианта цикла.	2		2	6	6	16
1.3	Специфика языка С. Массивы и указатели, процедуры и функции. Строки в С и функции работы со строками.	2		2	8	6	16
1.4	Основные алгоритмы и их сложность. Алгоритмы поиска, сортировки, поиска подстроки в строке.	2		2	8	6	20

1.5	Понятие рекурсии и рекурсивных алгоритмов. Алгоритмы перебора с возвратом, алгоритмы оптимального поиска.	3		3	8	6	20
1.6	Псевдослучайные числа. Алгоритмы генерации ПСЧ, проверки качества генераторов. Алгоритмы, использующие случайные числа.	3		3	8	6	20
Итого:		14		14	44	36	108

14. Методические указания для обучающихся по освоению дисциплины

Освоение дисциплины включает в себя лекционные занятия, лабораторные занятия и самостоятельную работу обучающихся. На первом занятии студент получает информацию для доступа к комплексу учебно-методических материалов.

Лекционные занятия посвящены рассмотрению теоретических основ дисциплины. Лабораторные занятия предназначены для формирования умений и навыков, закрепленных компетенциями по ОПОП. Самостоятельная работа студентов включает в себя проработку учебного материала лекций, разбор лабораторных заданий, подготовку к экзамену.

Для успешного освоения дисциплины рекомендуется подробно конспектировать лекционный материал, просматривать презентации (при наличии) по соответствующей теме, изучать основную и дополнительную литературу рекомендуемой библиографии,

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

а) основная литература:

№ п/п	Источник
1	Маер, А. В. Введение в структуры и алгоритмы обработки данных : учебное пособие / А. В. Маер, О. С. Черепанов. – Курган : КГУ, 2021. – 107 с. – ISBN 978-5-4217-0576-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/177907 . – Режим доступа: для авториз. пользователей.
2	Колмогорова, С. С. Основы искусственного интеллекта : учебное пособие для студентов / С. С. Колмогорова. – Санкт-Петербург : СПбГЛТУ, 2022. – 108 с. – ISBN 978-5-9239-1308-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/257804 . – Режим доступа: для авториз. пользователей.

б) дополнительная литература:

№ п/п	Источник
3	Защита программ и данных : учебное пособие. – Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2020 – Часть 1 : Способы анализа – 2020. – 72 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/180081 . – Режим доступа: для авториз. пользователей.
4	Защита программ и данных : учебное пособие. – Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2020 – Часть 2 : Способы защиты анализа – 2020. – 52 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/180082 . – Режим доступа: для авториз. пользователей.
5	Зариковская, Н. В. Технология программирования : учебное пособие / Н. В. Зариковская. – Москва : ТУСУР, 2018. – 130 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/313805 . – Режим доступа: для авториз. пользователей.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
6	Электронно-библиотечная система «Лань» - Режим доступа: https://e.lanbook.com
7	Электронный каталог Научной библиотеки Воронежского государственного университета. - Режим доступа: http://www.lib.vsu.ru .
8	Криптографические протоколы (10.05.01)/Степанец Ю.А. - Образовательный портал «Электронный университет ВГУ». — Режим доступа: https://edu.vsu.ru

16. Перечень учебно-методического обеспечения для самостоятельной работы

В качестве формы организации самостоятельной работы применяются методические указания для самостоятельного освоения и приобретения навыков работы со специализированным программным обеспечением. Самостоятельная работа

студентов: изучение теоретического материала; подготовка к лекциям, работа с учебно-методической литературой, подготовка отчетов по лабораторным работам, подготовка к экзамену.

Для обеспечения самостоятельной работы студентов в электронном курсе дисциплины на образовательном портале «Электронный университет ВГУ» сформирован учебно-методический комплекс, который включает в себя: программу курса, учебные пособия и справочные материалы, методические указания по выполнению заданий лабораторных работ. Студенты получают доступ к данным материалам на первом занятии по дисциплине.

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

Дисциплина реализуется с применением электронного обучения и дистанционных образовательных технологий. Для организации занятий рекомендован онлайн-курс «Б1.В.ДВ.05.01 Разработка приложений на C++ (10.05.01)», размещенный на платформе Электронного университета ВГУ (LMS moodle), а также Интернет-ресурсы, приведенные в п.15в.5.

18. Материально-техническое обеспечение дисциплины

Учебная аудитория для лекций: специализированная мебель, компьютер преподавателя, мультимедийный проектор, экран.

Учебная аудитория для лабораторных занятий: специализированная мебель, персональные компьютеры, мультимедийный проектор, экран, лабораторное оборудование программно-аппаратных средств обеспечения информационной безопасности.

Аудитория для самостоятельной работы: учебная мебель, компьютер с возможностью подключения к сети «Интернет» и электронной платформе Электронного университета ВГУ.

Программное обеспечение (см.файл МТО): ОС Windows v.7, 8, 10, набор утилит (архиваторы, файл-менеджеры), LibreOffice v.5-7, Foxit PDF Reader.

19. Оценочные средства для проведения текущей и промежуточной аттестаций

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

№ п/п	Наименования раздела дисциплины	Компетенция(и)	Индикатор(ы) достижения компетенции	Оценочные средства
1	Понятие алгоритма, алгоритмического мышления и информации. Понятие алгоритмической ошибки.	ПК-1	ПК-1.1	устный опрос, тест, лабораторная работа
2	Основные типы данных, операторы языка С. Основные функции стандартной библиотеки. Файлы. Понятие инварианта цикла.	ПК-1	ПК-1.2	устный опрос, тест, лабораторная работа
			ПК-1.3	устный опрос, тест, лабораторная работа
3	Специфика языка С. Массивы и указатели, процедуры и функции. Строки в С и функции работы со строками.	ПК-1	ПК-1.1	устный опрос, тест, лабораторная работа
4	Основные алгоритмы и их сложность. Алгоритмы поиска, сортировки, поиска подстроки в строке.	ПК-1	ПК-1.1	устный опрос, тест, лабораторная работа
			ПК-1.3	устный опрос, тест, лабораторная работа

5	Понятие рекурсии и рекурсивных алгоритмов. Алгоритмы перебора с возвратом, алгоритмы оптимального поиска.	ПК-2	ПК-2.3	устный опрос, тест, лабораторная работа
			ПК-2.3	устный опрос, тест, лабораторная работа устный опрос, тест, лабораторная работа
6	Псевдослучайные числа. Алгоритмы генерации ПСЧ, проверки качества генераторов. Алгоритмы, использующие случайные числа.	ПК-3	ПК-3.4	устный опрос, тест, лабораторная работа
			ПК-3.4	устный опрос, тест, лабораторная работа
Промежуточная аттестация, форма контроля - Экзамен				Перечень вопросов (КИМ№1)

20 Типовые оценочные средства и методические материалы, определяющие процедуры оценивания

20.1 Текущий контроль успеваемости

Контроль успеваемости по дисциплине осуществляется с помощью следующих оценочных средств:

- лабораторные работы.

Перечень лабораторных работ

1	Лабораторная работа 1.	Рекурсивный алгоритм «ханойские башни».
2	Лабораторная работа 2	Метод барьера на шахматной доске. Тестирование модульной программы с помощью автоматизированной системы.
3	Лабораторная работа 3	Реализация линейного конгруэнтного генератора.
4	Лабораторная работа 4	Реализация алгоритма оптимального поиска с помощью метода ветвей и границ (задача о коммивояжере и т.п.)
5	Лабораторная работа 5	Практическое выяснение параметров быстрой сортировки.
6	Лабораторная работа 6	Рекурсивная функция вычисления символа Якоби

Технология проведения

Все лабораторные работы обязательны для выполнения. Задание является общим для всех, выполняется индивидуально под наблюдением преподавателя.

Критерии оценивания

- оценивается «зачтено», если работа выполнена в полном объеме (приведены все задания, и они правильные, даны пояснения);
- оценивается «не зачтено», работа выполнена не полностью или в представленной части много ошибок

20.2 Промежуточная аттестация

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

Перечень вопросов к зачету (КИМ №1)

1. Жизненный цикл программ. Оценка трудоемкости этапов.
2. Жизненный цикл программ. Классификация программных средств как объектов разработки.
3. Жизненный цикл программ. Модели ЖЦП.
4. Оптимизация на уровне алгоритма.
5. Машинно-независимая оптимизация.
6. Машинно-зависимая оптимизация.
7. Оптимизация, выполняемая компиляторами.
8. Оптимизация на уровне ассемблера.

9. Мобильность и переносимость ПО. Стандарт ANSI C и его метрические ограничения.

10. Мобильность и переносимость ПО. Непереносимые конструкции языка C.
11. Отладка и тестирование. Методы тестирования и отладки.
12. Невозможность исчерпывающего тестирования.
13. Тестирование программ. Критерии выбора тестов.
14. Структурное и мутационное тестирование.
15. Стохастическое и функциональное тестирование.
16. Тестирование модулей.
17. Архитектура современных процессоров
18. Оптимизация, выполняемая современными процессорами.
19. Архитектура современных графических процессоров (GPU) и ее отличия от CPU
20. Модели программирования и памяти в GPU
21. Технология CUDA и OpenCL, их отличия
22. Базовые параллельные алгоритмы - параллельная редукция, сортировка.
23. Критерии оценки ответов на вопросы экзамена

Для оценивания результатов обучения используется - 4-балльная шкала:

«отлично», «хорошо», «удовлетворительно», «неудовлетворительно», критерии оценивания приведены ниже.

Оценка «отлично» - студент демонстрирует глубокое понимание темы, умеет распространять вытекающие из теории выводы.

Оценка «хорошо» - студент демонстрирует понимание теоретических положений темы и базовых понятий, но допускает неточности в ответах, испытывает затруднения в применении знаний к анализу состояния проекта.

Оценка «удовлетворительно» - студент отвечает не на все предложенные вопросы, но не менее, чем на половину из них; не демонстрирует способности применения теоретических знаний для анализа ситуаций.

Оценка «неудовлетворительно» - студент демонстрирует непонимание теоретических основ и базовых понятий курса.

Оценка промежуточной аттестации формируется как интегральная оценка по следующей формуле (При округлении оценки используется правило правильного округления. При получении оценки не менее 3 баллов, выставляется «зачтено», менее 3 баллов - «не зачтено». При этом, все лабораторные работы должны быть выполнены и защищены

$$Q_{\text{пром_ат}} = 0,2Q_{\text{KP1}} + 0,2Q_{\text{KP2}} + 0,6Q_{\text{экз}}$$

При округлении оценки используется правило правильного округления. При получении оценки не менее 3 баллов, выставляется «зачтено», менее 3 баллов - «не зачтено». При этом, все лабораторные работы должны быть выполнены и защищены.

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

ПК-1. Способен проводить анализ требований и выполнять работы по проектированию программных и аппаратных компонент системы безопасности компьютерных систем и сетей, в том числе с использованием современных методов и средств защиты информации;

ПК-2 Способен проводить исследования на всех этапах жизненного цикла программных средств в профессиональной деятельности

ПК-3 Способен осуществлять разработку, анализ и обосновывать эффективность применяемых математических методов защиты информации, возникающих при работе программных и программно-аппаратных средств защиты информации при решении профессиональных, исследовательских и прикладных задач

1. Выражения могут быть содержать

- а) только бинарные операции;
- б) только тернарные операции;
- в) только унарные или бинарными;
- г) унарные, бинарные или тернарными

2. Чему будет равно x

```
int x=1;  
x = ++x + ++x;  
cout << x << endl;
```

- а) 5
- б) 6
- в) 4

3. Чему будет равно x

```
int x=1;  
x = x++ + ++x;  
cout << x << endl;
```

- а) 5
- б) 6
- в) 4

4. Выберите логические операции (возможен множественный выбор)

- а) ~
- б) ||
- в) |
- г) &&
- д) ==

5. Выберите побитовые операции (возможен множественный выбор)

- а) +
- б) &
- в) ||
- г) !=
- д) <<

6. Чему будет равно z

```
uint8_t x=1, y=0, z=!((x&y)||y);
```

- а) 1
- б) 0
- в) 255

7. Чему будет равно z

```
uint8_t x=1, y=0, z=~((x&y)||y);
```

- а) 1
- б) 0
- в) 255

8. Что будет выведено в результате выполнения фрагмента кода

```
int i=4, j=0;  
do j+=++i; while(i<-3);  
cout << i << ' ' << j;
```

- а) 5 5
- б) 4 0
- в) 5 4

9. Что будет выведено в результате выполнения фрагмента кода

```
int a[]={2,2,2,5,5,3,5,4,2,1}, *p=a;  
while((p+=*p) < a+10) cout << ' ' << *p;
```

- а) 2 2 4
- б) 2 5 1
- в) 2 5 2

10. При вычислении выражений последовательность выполнения операций определяется

- а) только скобками;
- б) только приоритетом операций;
- в) скобками, приоритетом операций, а при одинаковом приоритете ассоциативностью операций

11. Выберите каким определениям функции может соответствовать вызов

f(3.14)

- а) void f(double x);

б) void f(float x);

в) void f(float x, float y);

г) void f(float x, float y=0);

д) float f(float x, float y=0);

е) void f(float x=0, float y);

12. Выберите корректный вариант освобождения памяти для динамического массива

int *buf= new char[20];

а) free(buf);

б) delete[] buf;

в) delete buf;

г) delete buf[];

13. Разница между формальными и фактическими параметрами функции заключается в том, что

а) первые определены в заголовке функции, а вторые - представляют собой значения, с которыми функция вызывается;

б) первые могут передаваться по значению, а вторые - по ссылке;

в) первые могут быть константными, а вторые - нет;

14. Классы необходимы для

а) обеспечения возможности взаимодействия кода из разных модулей приложения;

б) определения пользовательских типов данных;

в) объединения данных и операций над ними;

15. По умолчанию поля класса объявляются как

а) private

б) public

в) protected

16. Определение класса - это

а) вызов конструктора;

б) объявление всех его методов и полей;

в) инициализация всех его полей

17. Какая называется функция, не являющаяся методом класса, но имеющая доступ к его защищенным и внутренним членам

а) шаблонная;

б) статическая;

в) дружественная

18. Виртуальные функции - это

а) функции базового класса, которые не используются в производном классе;

б) функции базового класса, которые могут быть переопределены в производном классе

в) функции базового класса, которые не имеют реализации

19. Дружественная функция - это

а) функция другого класса, среди аргументов которой есть элементы данного класса

б) функция, являющаяся членом класса и объявленная с атрибутом friend;

в) функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса

20. Выберите утверждения, справедливые для абстрактных классов (возможен множественный выбор)

а) объект такого класса создать невозможно;

б) в таком классе нельзя определять методы;

в) класс, имеющий хотя бы один чисто виртуальный метод;

г) такой класс не может иметь наследников

21. Механизмы, через которые реализуется полиморфизм

а) перегрузка функций, наследование, шаблоны;

б) перегрузка функций, виртуальные функции, шаблоны;

в) наследование, виртуальные функции, дружественные функции

22. Можно ли в шаблоне класса определить статический метод?

а) да;

б) нет

23. Статические атрибуты класса (возможен множественный выбор)

а) существуют в единственном экземпляре, независимо от количества объектов;

б) не могут изменяться;

в) инициализируются при создании первого объекта;

г) инициализируются в начале выполнения программы

24. Выберите утверждения, которые можно считать справедливыми при использовании шаблонов (возможен множественный выбор)

а) повторное использование кода;

б) уменьшение времени компиляции;

в) возможность использования шаблонных реализаций интерфейсов при построении иерархий наследования;

г) упрощение анализа кода, содержащего шаблоны

25. Выберите корректный вариант описания шаблонной функции (возможен множественный выбор)

а) template <typename T> void f(T x, T y) { ... };

б) template <class T>; void f<T x, T y> { ... };

в) template <class T> void f(T& x, T& y) { ... };

г) template class T void f(T x, T x) (...);

26. Число параметров шаблона при определении шаблона функции определяется

а) числом параметризуемых типов;

б) числом аргументов функции

27. Выберите корректный вариант создания экземпляра объекта шаблонного класса

template <class T> class Vector

{

public:

Vector(int) { ... }

~ Vector () { ... }

}

а) Vector v(10);

б) Vector::<int> v(10);

в) Vector<int> v(10);

28. Что будет выведено в результате выполнения фрагмента кода

class A

{ public: void f() { cout << "A"; } };

class B : public A

{ public: void f() { cout << "B"; } };

B b;

A&a=b;

a.f();

а) BA

б) B

в) A

г) AB

29. Что обозначает объявление

extern int x;

а) переменная x не может быть изменена вне текущего модуля;

б) переменная x определена в другом исходном файле;

в) переменная x является статической

30. Этапы компиляции программы

а) препроцессинг → ассемблирование → компиляция → линковка

а) компиляция → препроцессинг → ассемблирование → линковка

а) компиляция → линковка → препроцессинг → ассемблирование

г) препроцессинг → компиляция → ассемблирование → линковка

Вопросы с коротким ответом

31. Распишите вывод для фрагмента кода

char *s="hmnmasoazermjrgtrzd",

*p[]={s+3,s+3,s+4,s+4,s+2}, **pp=p;

cout << s[2] << *p[3] << p[1][2] << pp[2][4];

32. Распишите вывод для фрагмента кода

int i=0, j=-1, k=-1;

if(++i || ++j && ++k) cout << i << ' ' << j << ' ' << k;

else cout << -i << ' ' << -j << ' ' << -k;

33. Распишите вывод для фрагмента кода

int i=-9, *p=&i;

cout << ++*p; cout << ' ' << *p++;

cout << ' ' << ++*--p << ' ' << i;

34. Распишите вывод для фрагмента кода

char *s="ldxjpxovjtoalmi";

int i,a[]={6,0,4,2,3,7,0,4,3,8};

for(i=0; i<10; i++) cout << *(s+(a+i));

35. Структура данных, представленная в виде группы ячеек одного типа, объединенных под единым

именем, применяемая для обработки большого количества однотипных данных

36. Особый тип переменной, хранящей адрес некоторой другой переменной

37. Один из способов реализации полиморфизма - возможность одновременного существования в одной области видимости нескольких различных вариантов применения оператора, имеющих одно и то же имя, но различающихся типами параметров, к которым они применяются

38. Функции, которые не являются членами класса, однако имеют доступ к его закрытым членам - полям и методам в секции private

39. Событие при выполнении программы, которое приводит к ее ненормальному или неправильному поведению (бывают аппаратные и программные)

- 1) 1. Выражения могут быть содержать
- 2) а) только бинарные операции;
- 3) б) только тернарные операции;
- 4) в) только унарные или бинарными;
- 5) г) унарные, бинарные или тернарными
- 6) 2. Чему будет равно x
- 7) int x=1;
- 8) x = ++x + ++x;
- 9) cout << x << endl;
- 10) а) 5
- 11) б) 6
- 12) в) 4
- 13) 3. Чему будет равно x
- 14) int x=1;
- 15) x = x++ + ++x;
- 16) cout << x << endl;
- 17) а) 5
- 18) б) 6
- 19) в) 4
- 20) 4. Выберите логические операции (возможен множественный выбор)
- 21) а) ~
- 22) б) ||
- 23) в) |
- 24) г) &&
- 25) д) ==
- 26) 5. Выберите побитовые операции (возможен множественный выбор)
- 27) а) +
- 28) б) &
- 29) в) ||
- 30) г) !=
- 31) д) <<
- 32) 6. Чему будет равно z
- 33) uint8_t x=1, y=0, z=!((x&& y)||y);
- 34) а) 1
- 35) б) 0
- 36) в) 255
- 37) 7. Чему будет равно z
- 38) uint8_t x=1, y=0, z=~((x&& y)||y);
- 39) а) 1
- 40) б) 0
- 41) в) 255
- 42) 8. Что будет выведено в результате выполнения фрагмента кода
- 43) int i=4, j=0;
- 44) do j+=++i; while(i<-3);
- 45) cout << i << ' ' << j;
- 46) а) 5 5
- 47) б) 4 0
- 48) в) 5 4
- 49) 9. Что будет выведено в результате выполнения фрагмента кода

- 50) int a[]={2,2,2,5,5,3,5,4,2,1}, *p=a;
51) while((p+=*p) < a+10) cout << ' ' << *p;
52) а) 2 2 4
53) б) 2 5 1
54) в) 2 5 2
55) 10. При вычислении выражений последовательность выполнения операций определяется
56) а) только скобками;
57) б) только приоритетом операций;
58) в) скобками, приоритетом операций, а при одинаковом приоритете ассоциативностью операций
59) 11. Выберите каким определениям функции может соответствовать вызов
60) f(3.14)
61) а) void f(double x);
62) б) void f(float x);
63) в) void f(float x, float y);
64) г) void f(float x, float y=0);
65) д) float f(float x, float y=0);
66) е) void f(float x=0, float y);
67) 12. Выберите корректный вариант освобождения памяти для динамического массива
68) int *buf= new char[20];
69) а) free(buf);
70) б) delete[] buf;
71) в) delete buf;
72) г) delete buf[];
73) 13. Разница между формальными и фактическими параметрами функции заключается в том, что
74) а) первые определены в заголовке функции, а вторые – представляют собой значения, с которыми функция вызывается;
75) б) первые могут передаваться по значению, а вторые – по ссылке;
76) в) первые могут быть константными, а вторые – нет;
77) 14. Классы необходимы для
78) а) обеспечения возможности взаимодействия кода из разных модулей приложения;
79) б) определения пользовательских типов данных;
80) в) объединения данных и операций над ними;
81) 15. По умолчанию поля класса объявляются как
82) а) private
83) б) public
84) в) protected
85) 16. Определение класса – это
86) а) вызов конструктора;
87) б) объявление всех его методов и полей;
88) в) инициализация всех его полей
89) 17. Какая называется функция, не являющаяся методом класса, но имеющая доступ к его защищенным и внутренним членам
90) а) шаблонная;
91) б) статическая;
92) в) дружественная
93) 18. Виртуальные функции – это
94) а) функции базового класса, которые не используются в производном классе;
95) б) функции базового класса, которые могут быть переопределены в производном классе
96) в) функции базового класса, которые не имеют реализации
97) 19. Дружественная функция – это
98) а) функция другого класса, среди аргументов которой есть элементы данного класса
99) б) функция, являющаяся членом класса и объявленная с атрибутом friend;
100) в) функция, объявленная в классе с атрибутом friend, но не являющаяся членом класса
101) 20. Выберите утверждения, справедливые для абстрактных классов (возможен множественный выбор)
102) а) объект такого класса создать невозможно;
103) б) в таком классе нельзя определять методы;
104) в) класс, имеющий хотя бы один чисто виртуальный метод;
105) г) такой класс не может иметь наследников
106) 21. Механизмы, через которые реализуется полиморфизм

- 107) а) перегрузка функций, наследование, шаблоны;
108) б) перегрузка функций, виртуальные функции, шаблоны;
109) в) наследование, виртуальные функции, дружественные функции
110) 22. Можно ли в шаблоне класса определить статический метод?
111) а) да;
112) б) нет
113) 23. Статические атрибуты класса (возможен множественный выбор)
114) а) существуют в единственном экземпляре, независимо от количества объектов;
115) б) не могут изменяться;
116) в) инициализируются при создании первого объекта;
117) г) инициализируются в начале выполнения программы
118) 24. Выберите утверждения, которые можно считать справедливыми при использовании шаблонов (возможен множественный выбор)
119) а) повторное использование кода;
120) б) уменьшение времени компиляции;
121) в) возможность использования шаблонных реализаций интерфейсов при построении иерархий наследования;
122) г) упрощение анализа кода, содержащего шаблоны
123) 25. Выберите корректный вариант описания шаблонной функции (возможен множественный выбор)
124) а) template <typename T> void f(T x, T y) { ... };
125) б) template <class T>; void f<T x, T y> { ... };
126) в) template <class T> void f(T& x, T& y) { ... };
127) г) template class T void f(T x, T x) (...);
128) 26. Число параметров шаблона при определении шаблона функции определяется
129) а) числом параметризуемых типов;
130) б) числом аргументов функции
131) 27. Выберите корректный вариант создания экземпляра объекта шаблонного класса
132) template <class T> class Vector
133) {
134) public:
135) Vector(int) { ... }
136) ~ Vector () { ... }
137) }
138) а) Vector v(10);
139) б) Vector::<int> v(10);
140) в) Vector<int> v(10);
141) 28. Что будет выведено в результате выполнения фрагмента кода
142) class A
143) { public: void f() { cout << "A"; } };
144) class B : public A
145) { public: void f() { cout << "B"; } };
146) B b;
147) A&a=b;
148) a.f();
149) а) BA
150) б) B
151) в) A
152) г) AB
153) 29. Что обозначает объявление
154) extern int x;
155) а) переменная x не может быть изменена вне текущего модуля;
156) б) переменная x определена в другом исходном файле;
157) в) переменная x является статической
158) 30. Этапы компиляции программы
159) а) препроцессинг → ассемблирование → компиляция → линковка
160) а) компиляция → препроцессинг → ассемблирование → линковка
161) а) компиляция → линковка → препроцессинг → ассемблирование
162) г) препроцессинг → компиляция → ассемблирование → линковка
163) Вопросы с коротким ответом

- 164) 31. Распишите вывод для фрагмента кода
165) char *s="hmnmasoazermjrgtrzd",
166) *p[]={s+3,s+3,s+4,s+4,s+2}, **pp=p;
167) cout << s[2] << *p[3] << p[1][2] << pp[2][4];
168) 32. Распишите вывод для фрагмента кода
169) int i=0, j=-1, k=-1;
170) if(++i || ++j && ++k) cout << i << ' ' << j << ' ' << k;
171) else cout << -i << ' ' << -j << ' ' << -k;
172) 33. Распишите вывод для фрагмента кода
173) int i=-9, *p=&i;
174) cout << ++*p; cout << ' ' << *p++;
175) cout << ' ' << ++*--p << ' ' << i;
176) 34. Распишите вывод для фрагмента кода
177) char *s="ldjxjpxovjtoalmi";
178) int i,a[]={6,0,4,2,3,7,0,4,3,8};
179) for(i=0; i<10; i++) cout << *(s+(a+i));
180) 35. Структура данных, представленная в виде группы ячеек одного типа, объединенных под
единым именем, применяемая для обработки большого количества однотипных данных
181) 36. Особый тип переменной, хранящей адрес некоторой другой переменной
182) 37. Один из способов реализации полиморфизма – возможность одновременного существования
в одной области видимости нескольких различных вариантов применения оператора, имеющих одно и
то же имя, но различающихся типами параметров, к которым они применяются
183) 38. Функции, которые не являются членами класса, однако имеют доступ к его закрытым членам
– полям и методам в секции private
184) 39. Событие при выполнении программы, которое приводит к ее ненормальному или
неправильному поведению (бывают аппаратные и программные)

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