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

УТВЕРЖДАЮ
заведующий кафедрой
кибербезопасности
информационных систем
С.Л. Кенин

22.03.2024

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Б1.В.08 Разработка безопасного программного обеспечения

1. Код и наименование направления подготовки/специальности:

10.05.01 Компьютерная безопасность

2. Профиль подготовки/специализация:

Математические методы защиты информации

3. Квалификация (степень) выпускника: Специалист

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины:

кибербезопасности информационных систем

6. Составители программы:

Бутузов Владимир Вячеславович, к.т.н., доцент кафедры кибербезопасности информационных систем

7. Рекомендована: НМС факультета ПММ, протокол № 5 от 22.03.2024
Внесены изменения: протокол УС факультета ПММ, протокол № 8 от 27.02.2024
Рекомендована с изменениями: протокол НМС факультета ПМ, протокол № 5 от 22.03.2024

8. Учебный год: 2024/2025 **Семестр(ы):** 7

9. Цели и задачи учебной дисциплины

Цель: изучение методов разработки безопасного программного обеспечения и методов контроля такой разработки.

Задачи:

Изучение содержания нормативных и методических документов, международных и национальных стандартов, передового опыта ведущих разработчиков программного обеспечения в области разработки безопасного программного обеспечения.

Изучение классов ошибок, допускаемых при разработке программного обеспечения.

Анализ потенциально опасных программных конструкций и алгоритмов, приводящих к нарушению конфиденциальности, целостности и доступности информации.

Изучение методов выявления уязвимостей в программном обеспечении.

Изучение методов устранения ошибок и уязвимостей в программном обеспечении.

Разработка безопасных алгоритмов программного обеспечения.

Изучение критериев и показателей оценки процессов разработки безопасного программного обеспечения.

Изучение методов контроля разработки безопасного программного обеспечения.

Разработка организационно-распорядительной и эксплуатационной документации, используемой в процессе разработки безопасного программного обеспечения.

Формирование описаний сведений об уязвимостях программного обеспечения и инцидентах информационной безопасности.

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 программных и аппаратных компонент системы безопасности компьютерных систем и сетей, в том числе с использованием современных методов и средств защиты информации	ПК-1.1	применяет современные методы разработки программного обеспечения и технологии программирования;	Знание: современные методы разработки программного обеспечения и технологии программирования; принципы комплексной разработки правил, процедур, приемов и методов, при создании средств защиты информации. Умеет: применять современные методы разработки программного обеспечения и технологии программирования; использовать современные математические методы и алгоритмы функционирования при создании компонентов программных средств защиты информации; использовать принципы комплексной разработки правил, процедур, приемов и методов, при создании средств защиты информации, в том числе с использованием
		ПК-1.2	использует современные математические методы и алгоритмы функционирования при создании компонентов программных средств защиты информации	

		ПК-1.3.	использует 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	современных методов и средств разработки программного обеспечения. Владеет принципалами комплексной разработки правил, процедур, приемов и методов, при создании средств защиты информации, в том числе с использованием современных методов и средств разработки программного обеспечения.
ПК-3	Способен участвовать в работах по проектированию систем защиты информации в компьютерных системах и сетях при решении профессиональных, исследовательских и прикладных задач	ПК-3.5.	выполняет проверку устойчивости приложений к внешнему несанкционированному доступу, в том числе проверка устойчивости веб-приложений к атакам, применение средств контроля безопасности, управление криптографическими средствами, а также организация мероприятий по обеспечению кибербезопасности	Умеет выполнять проверку устойчивости приложений к внешнему несанкционированному доступу, в том числе проверка устойчивости веб-приложений к атакам, применение средств контроля безопасности, управление криптографическими средствами, а также организация мероприятий по обеспечению кибербезопасности; участвовать в проектировании системы защиты информации и подсистем информационной безопасности компьютерной системы.
		ПК-3.6	способен участвовать в проектировании системы защиты информации и подсистем информационной безопасности компьютерной системы	

12. Объем дисциплины в зачетных единицах/час - 2/72.
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			
				7		
Аудиторные занятия	50			50		
в том числе: лекции	16			16		
Практические	0			0		
Лабораторные	34			34		
Самостоятельная работа	22			22		
Контроль	0			0		
Итого:	72			72		
Форма промежуточной аттестации	зачет с оценкой			зачет с оценкой		

13.1. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины	Реализация раздела дисциплины с помощью онлайн-курса, ЭУМК
1. Лекции			
1.1	Анализ программных реализаций	Основы тестирования черного ящика. Мониторинг: доступ к объектам ОС, вызов API ОС, сетевая активность. Основы статического анализа. Ассемблер. Архитектуры x86 и x64. Дизассемблер. Формат исполняемых файлов PE. Windows API. Приложения UEFI. Формат исполняемых файлов ELF. Linux API. Формат исполняемых файлов языков программирования с промежуточным кодом: Java, C#, Python. Декомпиляция. Идентификация абстракций языка C++ при декомпиляции. Виртуальные таблицы. Варианты передачи аргументов и возврата результата из подпрограмм. Динамический анализ. Отладка. Принципы функционирования отладчиков. Отладка в режиме пользователя и в режиме ядра. Снятие дампа. Пошаговая отладка. Программные и аппаратные точки останова. Методика динамического анализа: этапы и методы. Анализ потока данных. Наложение патчей на программы и процессы. Инструментализация. Анализ потока данных. Анализ потока управления. Противодействие антиотладке. Виртуализация. Эмуляция платформы для исследуемой программы. Изоляция исследуемой программы.	https://edu.vsu.ru/course/)
1.2	Защита программ от изучения.	Основы защиты от дизассемблирования. Обфусцирующие преобразования исходного кода. Обфусцирующие преобразования потока управления. Обфускация библиотечных вызовов, в том числе API ОС. Вызов библиотечных функций по их хеш-значениям. Обфусцирующие преобразования структур данных. Основы антиотладки. Упаковщики. Шифрование выполняемого кода. Самораспаковывающиеся архивы. Оверлеи. Полиморфный код. Техники обнаружения отладчика. Техники обнаружения гипервизора. Основы защиты от снятия дампа. Техники защиты от копирования.	
1.3	Вредоносное программное обеспечение	Классификация вредоносного ПО по модели поведения: наблюдатель, перехват и искажение. Виды вредоносного ПО. Вирусы и черви. Троянское ПО.	

		Rootkit, bootkit и backdoor. Вредоносные утилиты: эксплоиты, шеллкод, спуффинговое ПО, спам и флуд ПО, фишинговое ПО, рекламное ПО, программы-вымогатели, ПО для распространения запрещенной информации, DoS, ПО для создания и модификации вирусов. Бинарное вредоносное ПО на Windows. Бинарное вредоносное ПО на Linux. Вредоносное ПО на языках программирования с промежуточным кодом: Java, C#, Python. Вредоносное ПО на языках командных интерпретаторов: batch, PowerShell, bash. Вредоносные ПО в макросах документов. Вредоносное ПО на мобильных утройствах. Bootkit. Расположение в IPL, VBR, MBR, BIOS, UEFI. Secure Boot, Verified Boot, Measured Boot. Intel BootGuard. ARM Trusted Boot Board. Техники распространения вирусного вредоносного ПО. Техники скрытия вредоносного ПО от обнаружения.	
1.4	Внедрение вредоносного ПО.	Предпосылки к внедрению внедрению вредоносного ПО: уязвимости программного обеспечения, уязвимости политики безопасности, человеческий фактор. Методы внедрения вредоносного ПО: маскировка под «безобидное» ПО, подмена, прямое и косвенное ассоциирование.	
1.5	Противодействие вредоносному ПО.	Методы выявления вредоносного ПО: сигнатурное и эвристическое сканирование, контроль целостности, мониторинг информационных потоков, изолированная программная среда, программные ловушки. Принципы построения политики безопасности, обеспечивающей высокую защищенность от вредоносного ПО.	
2. Лабораторные работы			
2.1	Лабораторная работа №1.	Статический и динамический анализ программ на языках Java, C# и Python.	https://edu.vsu.ru/course/
2.2	Лабораторная работа №2.	Статический и динамический анализ программ для Android.	
2.3	Лабораторная работа №3.	Мониторинг активности процессов Windows и Linux.	
2.4	Лабораторная работа №4.	Статический анализ бинарных исполняемых файлов Windows.	
2.5	Лабораторная работа №5.	Статический анализ бинарных исполняемых файлов Linux.	
2.6	Лабораторная работа №6.	Отладка процессов в Windows.	
2.7	Лабораторная работа №7.	Отладка процессов в Linux.	
2.8	Лабораторная работа №8.	Обфускация исходного кода и его анализ.	
2.9	Лабораторная работа №9.	Защита исполняемых файлов от изучения.	
2.10	Лабораторная работа №10.	Изучение защищенных исполняемых файлов.	
2.11	Лабораторная работа №11.	Обнаружение вредоносного ПО с помощью ClamAV и YARA.	
2.12	Лабораторная работа №12.	Организация изолированной программной среды и аудита в Windows и Linux средствами по умолчанию	
2.13	Лабораторная работа №13.	Организация изолированной программной среды, аудита и доверенной загрузки Windows средствами отечественных средств защиты.	

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	Анализ программных реализаций	2	0	5	2	0	30
1.2	Защита программ от изучения.	4	0	8	5	0	40
1.3	Вредоносное программное обеспечение	4	0	7	6	0	38
1.4	Внедрение вредоносного ПО.	3	0	7	5	0	36
1.5	Противодействие вредоносному ПО.	3	0	7	4	0	36
	Итого:	16	0	34	22	0	72

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	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.
2	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.
3	Климентьев, К. Е. Компьютерные вирусы и антивирусы: взгляд программиста / К. Е. Климентьев. – Москва : ДМК Пресс, 2013. – 656 с. – ISBN 978-5-94074-885-4. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/63192. – Режим доступа: для авториз. пользователей.
4	Белоус, А. И. Программные и аппаратные трояны – способы внедрения и методы противодействия. Первая техническая энциклопедия : в 2 книгах / А. И. Белоус, В. А. Солодуха, С. В. Шведов ; под редакцией А. И. Белоуса. – Москва : Техносфера, 2019 – Книга 1 – 2019. – 688 с. – ISBN 978-5-94836-524-4. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/140565. – Режим доступа: для авториз. пользователей.

б) дополнительная литература:

№ п/п	Источник
5	Касперски, К. Фундаментальные основы хакерства. Искусство дизассемблирования / К. Касперски. – Москва : СОЛОН-Пресс, 2007. – 448 с. – ISBN 5-93455-175-2 . – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/13649. – Режим доступа: для авториз. пользователей.
6	Оголюк, А. А. Защита приложений от модификации. Дополнительные материалы : учебное пособие / А. А. Оголюк. – Санкт-Петербург : НИУ ИТМО, 2014. – 122 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/70849. – Режим доступа: для авториз. пользователей.
7	Штеренберг, С. И. Ассемблер в задачах защиты информации : учебное пособие / С. И. Штеренберг, А. В. Красов, В. Е. Радынская. – Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2019. – 82 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/180080. – Режим доступа: для авториз. пользователей.

в) информационные электронно-образовательные ресурсы**:**

№ п/п	Источник
8	Электронно-библиотечная система «Лань» - Режим доступа: https://e.lanbook.com
9	Электронный каталог Научной библиотеки Воронежского государственного университета. - Режим доступа: http//:www.lib.vsu.ru.

10	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 отчётов 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.08 Разработка безопасного программного обеспечения (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	Анализ программных реализаций	ПК-1	ПК-1.2	устный опрос, тест, лабораторная работа
			ПК-1.3	устный опрос, тест, лабораторная работа
		ПК-3	ПК-3.5	устный опрос, тест, лабораторная работа
			ПК-3.6	устный опрос, тест, лабораторная работа
2	Защита программ от	ПК-1	ПК-1.1	устный опрос, тест,

	изучения.			лабораторная работа
			ПК-1.2	устный опрос, тест, лабораторная работа
			ПК-1.3	устный опрос, тест, лабораторная работа
		ПК-3	ПК-3.5	устный опрос, тест, лабораторная работа
			ПК-3.6	устный опрос, тест, лабораторная работа
3	Вредоносное программное обеспечение	ПК-1	ПК-1.3	устный опрос, тест, лабораторная работа
		ПК-3	ПК-3.5	устный опрос, тест, лабораторная работа
			ПК-3.6	устный опрос, тест, лабораторная работа
4	Внедрение вредоносного ПО.	ПК-1	ПК-1.1	устный опрос, тест, лабораторная работа
			ПК-1.2	устный опрос, тест, лабораторная работа
5	Противодействие вредоносному ПО.	ПК-1	ПК-1.1	устный опрос, тест, лабораторная работа
			ПК-1.2	устный опрос, тест, лабораторная работа
			ПК-1.3	устный опрос, тест, лабораторная работа
		ПК-3	ПК-3.5	устный опрос, тест, лабораторная работа
			ПК-3.6	устный опрос, тест, лабораторная работа
Промежуточная аттестация, форма контроля - зачет с оценкой				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.	Статический и динамический анализ программ на языках Java, C# и Python.
2	Лабораторная работа №2.	Статический и динамический анализ программ для Android.
3	Лабораторная работа №3.	Мониторинг активности процессов Windows и Linux.
4	Лабораторная работа №4.	Статический анализ бинарных исполняемых файлов Windows.
5	Лабораторная работа №5.	Статический анализ бинарных исполняемых файлов Linux.
6	Лабораторная работа №6.	Отладка процессов в Windows.
7	Лабораторная работа №7.	Отладка процессов в Linux.
8	Лабораторная работа №8.	Обфускация исходного кода и его анализ.
9	Лабораторная работа №9.	Защита исполняемых файлов от изучения.
10	Лабораторная работа №10.	Изучение защищенных исполняемых файлов.
11	Лабораторная работа №11.	Обнаружение вредоносного ПО с помощью ClamAV и YARA.

12	Лабораторная работа №12.	Организация изолированной программной среды и аудита в Windows и Linux средствами по умолчанию
13	Лабораторная работа №13.	Организация изолированной программной среды, аудита и доверенной загрузки Windows средствами отечественных средств защиты.

Технология проведения
Все лабораторные работы обязательны для выполнение. 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 экзамену.

Перечень вопросов к экзамену (КИМ №1)
1. Анализ программных реализации черного ящика. Мониторинг доступов к объектам ОС, вызовов API ОС, сетевой активности.
2. Статический метод анализа программных реализаций.
3. Декомпиляция бинарных файлов в язык C++.
4. Динамический метод анализа программных реализаций.
5. Анализ потока данных при помощи инструментализации.
6. Анализ потока управления при помощи инструментализации.
7. Противодействие антиотладке при помощи инструментализации.
8. Виртуализация. Эмуляция платформы для исследуемой программы. Изоляция исследуемой программы.
9. Защита программ от дизассемблирования.
10. Обфусцирующие преобразования исходного кода.
11. Обфусцирующие преобразования потока управления. Обфускация библиотечных вызовов, в том числе API ОС.
12. Обфусцирующие преобразования структур данных.
13. Особенности анализа обфусцированных программ.
14. Защита программ от отладки.
15. Упаковщики. Шифрование выполняемого кода. Самораспаковывающиеся архивы. Оверлеи. Полиморфный код.
16. Техники обнаружения отладчика.
17. Техники обнаружения гипервизора.
18. Основы защиты от снятия дампа.
19. Техники защиты от копирования.
20. Особенности анализа программ, защищенных от отладки.
21. Классификация вредоносного ПО по модели поведения: наблюдатель, перехват и искажение.
22. Виды вредоносного ПО. Вирусы и черви. Троянское ПО. Rootkit, bootkit и backdoor. Вредоносные утилиты: эксплоиты, шеллкод, спуффинговое ПО, спам и флуд ПО, фишинговое ПО, рекламное ПО, программы-вымогатели, ПО для распространения запрещенной информации, DoS, ПО для создания и модификации вирусов.
23. Бинарное вредоносное ПО на Windows.
24. Бинарное вредоносное ПО на Linux.
25. Вредоносное ПО на языках программирования с промежуточным кодом: Java, C#, Python.
26. Вредоносное ПО на языках командных интерпретаторов: batch, PowerShell, bash.

27. Вредоносные ПО в макросах документов.
28. Вредоносное ПО на мобильных утройствах.
29. Bootkit. Расположение в IPL, VBR, MBR, BIOS, UEFI. Secure Boot, Verified Boot, Measured Boot. Intel BootGuard. ARM Trusted Boot Board.
30. Техники распространения вирусного вредоносного ПО.
31. Техники скрытия вредоносного ПО от обнаружения.
32. Предпосылки к внедрению программных закладок: уязвимости программного обеспечения, уязвимости политики безопасности, человеческий фактор.
33. Методы внедрения программных закладок: маскировка под безобидное программное обеспечение, подмена, прямое и косвенное ассоциирование.
34. Принципы построения политики безопасности, обеспечивающей высокую защищенность от программных закладок.
35. Методы выявления программных закладок: сигнатурное и эвристическое сканирование. Инструмент YARA.
36. Контроль целостности программного обеспечения, как метод выявления программных закладок.
37. Контроль целостности конфигурации защищаемой системы, как метод выявления программных закладок.
38. Мониторинг информационных потоков, как метод выявления программных закладок.
39. Методы выявления программных закладок, изолированная программная среда, программные ловушки.
40. Построение изолированной программной среды с помощью Windows AppLocker.
41. Построение изолированной программной среды с помощью SELinux.
42. Построение изолированной программной среды на Windows с помощью ПО DallasLock.
43. Защита процесса загрузки ОС с помощью отечественных АМДЗ.
44. Аудит в ОС Windows.
45. Аудит в ОС Linux.
46. Мероприятия по организационному сопровождению антивирусной защиты.

Критерии оценки ответов на вопросы зачета с оценкой

Для оценивания результатов обучения на зачете с оценкой используется – 4-балльная шала:
«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

$$Q_{пром\_ат} = 0{,}2Q_{КР1} + 0{,}2Q_{КР2} + 0{,}6Q_{экз}$$

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;

1) закрытые задания (тестовые, средний уровень сложности):

1. В чем состоят основные этапы программирования?
   - **создание спецификации задачи, разработка алгоритма, разработка и проверка правильности работы программы;**
   - разработка алгоритма, разработка программы, тестирование программы;
   - разработка алгоритма, разработка программы, тестирование и отладка программы.
2. Что понимают под алгоритмом решения задачи?
   - последовательность инструкций, выполняемых процессором при решении задачи;
   - набор инструкций, определяющих действия процессора при решении задачи;
   - **конечный набор инструкций, определяющих действия процессора при решении задачи.**
3. Какие требования предъявляют к алгоритму?
   - дискретность элементарность действий, детерминированность, корректность, результативность, массовость;
   - **дискретность элементарность действий, детерминированность, конечность, результативность, массовость;**
   - дискретность элементарность действий, детерминированность, конечность, эффективность, массовость;
4. Какой набор инструкций является минимально необходимым для описания алгоритмов?
   - именование значений, помеченное действие, переход к помеченному действию, составное действие.
   - именование значений, помеченное действие, условное действие, циклическое действие, составное действие.
   - **именование значений, условное действие, циклическое действие, составное действие.**
5. Что понимается под тестированием программы?
   - этап отладки, на котором выявляются синтаксические ошибки;
   - **этап отладки, на котором выявляются содержательные ошибки;**
   - этап проверки корректности алгоритма, реализуемого программой.
6. Какой формат имеют двухадресные команды?
   - код операции, адрес операнда;
   - адрес операнда, адрес операнда;
   - **код операции, адрес операнда, адрес операнда.**
7. Какое место в массиве резервируется для искомого элемента при его поиске методом барьерного элемента?
   - центральное место;
   - **последнее место;**
   - предпоследнее место.
8. В каком случае делается наибольшее число сравнений при пузырьковой сортировке *n*-элементного массива?
   - **наименьший элемент стоит на последнем месте, независимо от положения наибольшего;**
   - наибольший элемент стоит на первом месте, независимо от положения наименьшего;
   - наибольший элемент предшествует наименьшему, а наименьший находится в центре
   - массива.
9. В каком случае делается наибольшее число сравнений при сортировке вставками *n*-элементного массива?

- наименьший элемент стоит на последнем месте, независимо от положения наибольшего;
- **массив отсортирован в обратном порядке;**
- наибольший элемент стоит на первом месте, независимо от положения наименьшего.

10. В программе описано класс и объект class A {public: int a, b, c; }; A * obj; Как обратиться к атрибуту c?

A. obj.c
B. obj→ c
C. obj A → → c
D. obj→ A.c

11. Свойство формы name – это…

A.Имя формы, используется для управления формой и доступа к компонентам формы
B. Текст заголовка (надпись на форме)
C. Значок в заголовке диалогового окна, обозначающий кнопку вывода системного меню

12. Что, помимо полей, могут включать в себя структуры?

A. Указатель на ту же структуру
B. Заголовки функций
C. Еще одну структуру
D. Ничего

13. Функция добавляет s2 к s1 и возвращает s1.

A. strcat(s1, s2)
B. strcmp(s1,s2)
C. strcpy(s1, s2)

14. Инкремент перемещает указатель к следующему элементу массива, значение указателя изменяется на

A. на единицу
B. на величину sizeof (тип)
C. величину sizeof (тип) * размер массива

15. Правильно ли, что «Структура является собранием одной или нескольких переменных».

A. Да
B. Нет
C. Только нескольких

16. Чтобы обратиться к искомому элементу входящего в структуру массива нужно:

A. Указать адрес массива
B. Указать индекс
C. Сделать все выше перечисленное

17. Первым символом идентификатора НЕ может быть

A. буква
B. знак подчеркивания
C. цифра

18. Существует четыре спецификатора типа, уточняющих внутреннее представление и диа-пазон значений стандартных типов:

A. short, long, signed , unsigned
B. const, long, char , unsigned
C. double, int, signed , unsigned

19. Вызовет данный код ошибку компиляции? class Rectangle { public: int a, b; int sum (); int square (); ~ Rect (); };

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

20. Укажите правильное использование оператора friend.

A. class A {int_friend CountPass (); private: short i;};
B. class A {public: friend int H :: CountPass (); private: short i;}; +
C. class A {public: int A1 :: CountPass (); friend: short i;};
D. class A {public: friend int H :: q; short i;};

21. Принцип объектно-ориентированного программирования, заключающийся в объедине-нии атрибутов и методов объекта с целью обеспечения сохранности данных, называется:

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

22. Принцип объектно-ориентированного программирования, заключающийся пере-даче свойств и методов базового класса классу наследнику, называется:
A. Наследование. +
B. Воспроизведение.
C. Преобразование.
D. Делегирование.

23. Величины типа **double** обычно занимают приведенное количество байтов
A. 4
B. 8 +
C. 10

24. Указатели чаще всего используют при работе
A. с динамической памятью+
B. с массивами
C. со структурами

25. Простейшее объявление указателя на объект имеет вид:
A. тип *имя;+
B. тип ^имя;
C. тип &имя;

26. Объявление строки является корректным:
A. char str[8] = ‘One text’;
B. char str(6) = ‘My text’;
C. char str[10] = “New text”;+
D. char str(7) = “Text”;

27. За каждый проход элементы последовательно сравниваются попарно и, если порядок в паре неверный выполняется обмен элементов. Так работает алгоритм сортировки
A. методом вставки
B. методом пузырька+
C. методом поиска минимума (максимума)

ПК-3. Способен участвовать в работах по проектированию систем защиты информации в компьютерных системах и сетях при решении профессиональных, исследовательских и прикладных задач

1) закрытые задания (тестовые, средний уровень сложности):

1. Какая функция, не будучи компонентом класса, имеет доступ к его защищенным и внутренних компонентов?
A. Шаблонная.
B. Полиморфная.
C. Дружественная. +
D. Статическая.

2. Один из трех примеров применения условного оператора является некорректным, какой?
A. if (a<0) b = 1;
B. if (a++) b++;
C. if (b>a) then max = b else max = a; +

3. Последовательность операторов, приведенная ниже, используется
```
for (int i=1; i<n; i++)
if (max<a[i])
{
max=a[i];
max_i=i;
}
```
A. сортировки массива
B. для нахождения максимума +
C. вывода на экран элементов массива

4. Пример инициализации многомерного массива является некорректным
A. int mas2 [][]={ {1, 1}, {0, 2}, {1, 0} };
B. int mas2 [3][2]={1, 1, 0, 2, 1, 0};
C. int mas2 [3][2]={1, 1, 0};

D. int mas2 [3,2]={1, 1, 0, 2, 1, 0}; +

5. Последовательность операторов, приведенная ниже, используется для

```
for (int i=0; i<n; i++)
{
s[i]=0;
for (int j=0; j<m; j++)
s[i]+=x[i][j];
}
```

A. сортировки массива

B. вывода на экран элементов массива

C. нахождения суммы элементов массива +

6. Какие существуют особенности документа для описания тестовых процедур?

(1) содержат описание последовательности действий, необходимых для выполнения тестового набора

(2) процедуры должны быть сформулированы так, чтобы их мог выполнить инженер, незнакомый с данным проектом

(3) процедуры для автоматизированных тестов должны содержать только информацию для запуска и анализа результатов

(4) процедуры автоматически выполняют тестовые наборы

7. Можно ли гарантировать безопасность метода регрессионного тестирования в условиях отсутствия информации об изменениях в программе?

(1) нет

(2) да

8. При создании очередной версии программы была добавлена функция A, функция D была удалена, функция C – изменена, а функция U – оставлена без изменений. К какой группе относится тест, покрывающий только функцию A?

(1) тесты, пригодные для повторного использования

(2) тесты, требующие повторного запуска

(3) устаревшие тесты

(4) новые тесты

9. Какими преимуществами обладает методика уменьшения объема тестируемой программы?

(1) уменьшается время компиляции тестируемой программы

(2) уменьшается время выполнения тестируемой программы

(3) уменьшается время работы метода отбора тестов

(4) уменьшается риск пропуска ошибки

10. На предыдущей версии программы тест 1 завершился в состоянии A, тест 2 – в состоянии B, а тест 3 – в состоянии C. На текущей версии программы тест 1 завершился в состоянии A, тест 2 – в состоянии C, а тест 3 – в состоянии D. На базе какого теста наиболее целесообразна разработка новых тестов?

(1) 1

(2) 2

(3) 3

11. Является ли программа аналогом математической формулы?

(1) да

(2) нет

(3) математические формулы и программы не сводятся друг к другу

12. Какие предъявляются требования к идеальному критерию тестирования?

(1) достаточность

(2) достижимость

(3) полнота

(4) проверяемость

13. Для хранения значения переменной объединения выделяется памяти...

A. Столько, чтобы вместить самый "широкий" элемент объединения; +

B. Сумма значений занимаемых каждым элементом в отдельности;

C. Столько, чтобы вместить первый элемент объединения.

14. Компонент, предоставляющий возможность ввода данных в поле редактирования путем набора на клавиатуре или выбором из списка

A.Button

B.Memo

C. CheckBox

D. ComboBox +

15. Элемент управления: timage

A. Отображает графическое изображение на форме +

B. Служит для отображения простейших графических объектов на форме: окружность, квадрат и т.п.

C. Позволяет создать на форме прокручиваемую область с размерами большими, чем экран, на которой можно разместить объекты

16. Компоненты, которые видны во время работы приложения, с ними напрямую может взаимодействовать пользователь, называются…

A. Визуальными +

B. Невизуальными

C. Активными

17. Деструктором называется?

1.метод, который уничтожает объект;

2.метод, который удаляет объект;

3.метод, который освобождает память, занимаемую объектом;

4.системная функция, которая освобождает память, занимаемую объектом;

18. Что называется наследованием?

1. это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы;

2. это механизм переопределения методов базового класса;

3. это механизм, посредством которого производный класс получает все поля базового класса;

4. это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить;

19. Приведен следующий фрагмент консольного приложения:

```
class MyClass
{
int m_age;
MyClass(int age):m_age(age){
}
};
int main()
{
MyClass(21);
return 0;
}
```

Ошибка в программном коде вызвана:

1. Отсутствуют данные в теле конструктора

2. Для конструктора не задан спецификатор доступа public

3. Неправильное присвоение данных в целочисленную переменную

4. Нельзя передавать в конструктор значение в формате десятичного числа

20.

```
class A
{
public:
A(){}
~A(){}
};
```

В классе объявлены:

1. Конструктор и деструктор

2. Два пустых конструктора

3. Два абстрактных метода

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