

МИНОБРАЗОВАНИЯ 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.О.56.01 Инженерия программного
обеспечения

1. Код и наименование направления подготовки/специальности:

10.05.01 Компьютерная безопасность

2. Профиль подготовки/специализация:

Безопасность компьютерных систем и сетей

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

4. Форма обучения: очная

5. Кафедра, отвечающая за реализацию дисциплины:

кибербезопасности информационных систем

6. Составители программы:

Иванкин Михаил Петрович, к.т.н., доцент кафедры кибербезопасности информационных систем

7. Рекомендована:

НМС факультета ПММ, протокол № 7 от 26.05.2023г.

8. Учебный год: 2029/2030

Семестр(ы): 9

9. Цели и задачи учебной дисциплины

Целью и задачами дисциплины являются практическое введение студентов в технологии проектирования прикладных программных систем с использованием унифицированных языков визуального проектирования прикладных программных систем и технологии Framework разработки приложений масштаба предприятия. Подготовка студентов к работе по созданию программного обеспечения в проектных группах. Обучение методам командной работы.

10. Место учебной дисциплины в структуре ОПОП: дисциплина относится к обязательной 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
ОПК-4.1	Способен организовывать защиту информации в компьютерных системах и сетях (по областям применения).	ОПК-4.1.3	Способен использовать языки и системы программирования, инструментальные средства при обеспечении защиты информации в компьютерных системах при решении различных профессиональных, исследовательских и прикладных задач.	Умеет: использовать языки и системы программирования, инструментальные средства при обеспечении защиты информации в компьютерных системах при решении различных профессиональных, исследовательских и прикладных задач. Владение языками и системами программирования, инструментальными средствами при обеспечении защиты информации в компьютерных системах при решении различных профессиональных, исследовательских и прикладных задач.
ОПК-4.3	Способен разрабатывать и анализировать корректность политики информационной безопасности компьютерных систем и сетей (по областям применения).	ОПК-4.3.5	Способен применять программные средства прикладного, системного и специального назначения при разработке и анализе политики информационной безопасности.	Умеет: применять программные средства прикладного, системного и специального назначения при разработке и анализе политики информационной безопасности. Владение программными средствами прикладного, системного и специального назначения при разработке и анализе политики информационной безопасности.

12. Объем дисциплины в зачетных единицах/час - 5/180.

Форма промежуточной аттестации - зачет оценкой.

13. Трудоемкость по видам учебной работы

Вид учебной работы	Трудоёмкость (часы)				
	Всего	В том числе в интерактивной форме	По семестрам		
			9		
Аудиторные занятия	108		108		
в том числе: лекции	36		36		
Практические	0		0		
Лабораторные	72		72		
Самостоятельная работа	72		72		
Контроль	0		0		
Итого:	180		180		
Форма промежуточной аттестации	зачет оценкой		зачет 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	Жизненный цикл программного обеспечения	Программные системы. Основные проблемы разработки сложных программных систем. Жизненный цикл ПО. Стандартизация жизненного цикла в системе государственных стандартов ЕСПД и ГОСТ Р ИСО/МЭК 12207. Модели жизненного цикла (каскадная, спиральная, формальные и др.). Гибкие методологии разработки (XP, Scrum и др.).	https://edu.vsu.ru/course/
1.2	Качество программных систем	Критерии оценки качества программных систем, характеристики качества и метрики качества. Методы контроля качества. Модель процесса оценивания качества. ГОСТ Р ИСО 9000 и ГОСТ Р ИСО 9126. Инструментальные системы оценки качества программных систем.	
1.3	Анализ и разработка требований	Функциональные и нефункциональные требования. Разработка требований. Спецификация требований к программному обеспечению, техническое задание (ГОСТ 19.201-78, ГОСТ 34.602-89). Методы первичного сбора требований. Модели системы. Языки спецификации требований (сценарии, PDL, потоковые диаграммы и др.). Оценка требований (ГОСТ Р ИСО/МЭК 12207). Методы аттестации требований (обзор, протипирование, тестовые сценарии). Управление требованиями. Инструментальные средства отслеживания и контроля требований.	
1.4	Проектирование архитектуры программных систем	Архитектурное и детальное проектирование. Основные принципы проектирования. Сквозная функциональность. Сцепление и связность. Архитектурные стили (парадигмы: объектно-ориентированная, компонентная, многослойная, DDD, клиент-сервер, многоуровневая, SOA и др.). Принципы объектно-ориентированного проектирования (SOLID). Паттерны проектирования. Унифицированный язык моделирования (UML). Документирование архитектуры.	

1.5	Аттестация и верификация	Понятие аттестации и верификации. Методы инспектирования. Инструментальные средства инспектирования и оценки качества. Тестирование ПО. Неразрешимость проблемы тестирования. Виды и уровни тестирования. Стратегии восходящего и нисходящего тестирования. Методы «белого» и «черного» ящика. Автоматизированное и ручное тестирование. Разработка через тестирование (TDD). Непрерывная интеграция. Покрывание кода тестами. Методы отладки программ. Инструментальные средства поддержки тестирования и отладки.	
2. Лабораторные работы			
2.1	Лабораторная работа №1. "Методы программирования"	В языке C# список (List<T>) реализован на основе массива. Необходимо написать класс, который будет иметь такую же функциональность, как и List<T>, используя технологию Test-Driven Development (TDD).	
2.2	Лабораторная работа №2. "Методы программирования" Применение стандартных контейнеров данных для решения задач	Необходимо найти первые 10 часто встречающихся слов из текста объемом в 3-5 Мб. Напишите три метода, решающих эту задачу, причем в одном методе в качестве контейнера для данных используется SortedDictionary<TKey, TValue>, в другом -- Dictionary<TKey, TValue>, в третьем -- SortedList<TKey, TValue>. Сравните время работы для каждого из методов.	
2.3	Лабораторная работа №3. "Методы программирования"	Напишите класс AVLTree<T>, который содержит методы удаления элемента из дерева, поиска элемента в дереве, вставки элемента, балансировки. Напишите консольное приложение, которое - генерирует массив из случайных 10000 чисел; - создает АВЛ-дерево из чисел этого массива; - удаляет из дерева числа, которые находились в массиве на месте с 5000 до 7000; - выполняет поиск каждого элемента массива и замеряет общее время, затраченное на поиск; - создайте SortedDictionary<int, int> на основе чисел массива. Также удалите из этого контейнера числа, которые находились в массиве на месте с 5000 до 7000 и замерьте время, затраченное на поиск всех элементов массива в SortedDictionary. Сравните время работы двух коллекций.	
2.4	Лабораторная работа №4. "Методы программирования"	1. Напишите класс HashTable<TKey, TValue>, который содержит открытые методы: удаления элемента, поиска элемента по ключу, вставки элемента, реализует интерфейс IEnumerable <KeyValuePair< TKey, TValue>> и необходимые внутренние методы(например, увеличение таблицы). Разрешение коллизий: методом цепочек либо открытая адресация(линейное, квадратичное или двойное исследование). 2. Напишите юнит-тесты, которые проверяют работу методов класса хеш-таблицы. 3. Напишите консольное приложение, которое замеряет время работы класса Dictionary и написанного Вами. Для этого нужно написать два метода. На входе для каждого метода - массив слов, полученных из файла WarAndWorld.txt В методе создается частотный словарь для слов	https://edu.vsu.ru/course/

		данного файла. Из него получить выборку слов длины 7. И каждое из слов выборки удалить из основного словаря. Сравните время работы двух коллекций.	
2.5	Лабораторная работа №5. "Методы программирования". Реализация и использование списка с пропусками	Пример задания: допишите метод удаления элемента (поиска элемента и т.п. - по вариантам) в проект, созданный на лекции. Сравните время работы по поиску, вставке, удалению элементов разных типов двух коллекций --- стандартного SortedList и списка с пропусками на не менее чем 10000 элементов.	
2.6	Лабораторная работа №6. "Методы программирования". Использование очереди с приоритетами (пирамиды, кучи) для решения различных задач	Напишите приложение, решающее задачу (по вариантам) и использующее описанный Вами класс. Пример задачи: Дано k отсортированных списков с общим количеством n элементов. Надо их соединить в один отсортированный список за время $O(n \log k)$, используя пирамиду.	

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	Жизненный цикл программного обеспечения	6		6	14	0	26
1.2	Качество программных систем	6		14	14	0	34
1.3	Анализ и разработка требований	8		18	14	0	40
1.4	Проектирование архитектуры программных систем	8		18	14	0	40
1.5	Аттестация и верификация	8		16	16	0	40
Итого:		36		72	72	0	180

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	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.
2	Технология программирования CUDA : учебное пособие / Д. Н. Тумаков, Д. Е. Чикрин, А. А. Егорчев, С. В. Голоусов. – Казань : КФУ, 2017. – 112 с. – ISBN 978-5-00019-913-8. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/130543 . – Режим доступа: для авториз. пользователей.

б) дополнительная литература:

№ п/п	Источник
3	Асташова, Т. А. Основы программирования : учебное пособие / Т. А. Асташова. – Новосибирск : НГТУ, 2022. – 92 с. – ISBN 978-5-7782-4843-4. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/404750 . – Режим доступа: для авториз. пользователей.
4	Информационная безопасность и защита информации. Том 2 : сборник научных трудов / А. И. Астайкин, А. П. Мартынов, Д. Б. Николаев, В. Н. Фомченко ; составители А. И. Астайкин [и др.]. – Саров : РФЯЦ-ВНИИЭФ, 2017 – Том 2 – 2017. – 500 с. – ISBN 978-5-9515-0299-5. – Текст : электронный // Лань : электронно-библиотечная система. – URL: https://e.lanbook.com/book/243458 . – Режим доступа: для авториз. пользователей.

в) информационные электронно-образовательные ресурсы:

№ п/п	Источник
5	Электронно-библиотечная система «Лань» - Режим доступа: https://e.lanbook.com
6	Электронный каталог Научной библиотеки Воронежского государственного университета. - Режим доступа: http://www.lib.vsu.ru .
7	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.О.57.01 Инженерия 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	Жизненный цикл программного обеспечения	ОПК-4.1	ОПК-4.1.3	устный опрос, тест, лабораторная работа
2	Качество программных систем	ОПК-4.1	ОПК-4.1.3	устный опрос, тест, лабораторная работа
		ОПК-4.3	ОПК-4.3.5	устный опрос, тест, лабораторная работа
3	Анализ и разработка требований	ОПК-4.1	ОПК-4.1.3	устный опрос, тест, лабораторная работа
		ОПК-4.3	ОПК-4.3.5	устный опрос, тест, лабораторная работа
4	Проектирование архитектуры программных систем	ОПК-4.1	ОПК-4.1.3	устный опрос, тест, лабораторная работа
		ОПК-4.3	ОПК-4.3.5	устный опрос, тест, лабораторная работа
5	Аттестация и верификация	ОПК-4.3	ОПК-4.3.5	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:

- контрольные работы,
- лабораторные работы.

Перечень лабораторных работ

1	Лабораторная работа №1. "Методы программирования"	В языке C# список (List<T>) реализован на основе массива. Необходимо написать класс, который будет иметь такую же функциональность, как и List<T>, используя технологию Test- Driven Development (TDD).
2	Лабораторная работа №2. "Методы программирования" Применение стандартных контейнеров данных для решения задач	Необходимо найти первые 10 часто встречающихся слов из текста объемом в 3-5 Мб. Напишите три метода, решающих эту задачу, причем в одном методе в качестве контейнера для данных используется SortedDictionary<TKey, TValue>, в другом -- Dictionary<TKey, TValue>, в третьем -- SortedList<TKey, TValue>. Сравните время работы для каждого из методов.
3	Лабораторная работа №3. "Методы программирования"	Напишите класс AVLTree<T>, который содержит методы удаления элемента из дерева, поиска элемента в дереве, вставки элемента, балансировки. Напишите консольное приложение, которое - генерирует массив из случайных 10000 чисел; - создает АВЛ-дерево из чисел этого массива; - удаляет из дерева числа, которые находились в массиве на месте с 5000 до 7000; - выполняет поиск каждого элемента массива и замеряет общее время, затраченное на поиск; - создайте SortedDictionary<int, int> на основе чисел массива. Также удалите из этого контейнера числа, которые находились в массиве на месте с 5000 до 7000 и замерьте время, затраченное на поиск всех элементов массива в SortedDictionary. Сравните время работы двух коллекций.
4	Лабораторная работа №4. "Методы программирования"	1. Напишите класс HashTable<TKey, TValue>, который содержит открытые методы: удаления элемента, поиска элемента по ключу, вставки элемента, реализует интерфейс IEnumerable <KeyValuePair< TKey, TValue>> и необходимые внутренние методы(например, увеличение таблицы). Разрешение коллизий: методом цепочек либо открытая адресация(линейное, квадратичное или двойное исследование). 2. Напишите юнит-тесты, которые проверяют работу методов класса хеш-таблицы. 3. Напишите консольное приложение, которое замеряет время работы класса Dictionary и написанного Вами. Для этого нужно написать два метода. На входе для каждого метода - массив слов, полученных из файла WarAndWorld.txt В методе создается частотный словарь для слов данного файла. Из него получить выборку слов длины 7. И каждое из слов выборки удалить из основного словаря. Сравните время работы двух коллекций.
5	Лабораторная работа №5. "Методы программирования". Реализация и использование списка с пропусками	Пример задания: допишите метод удаления элемента (поиска элемента и т.п. - по вариантам) в проект, созданный на лекции. Сравните время работы по поиску, вставке, удалению элементов разных типов двух коллекций --- стандартного SortedList и списка с пропусками на не менее чем 10000 элементов.
6	Лабораторная работа №6. "Методы программирования". Использование очереди с приоритетами (пирамиды, кучи) для решения различных задач	Напишите приложение, решающее задачу (по вариантам) и использующее описанный Вами класс. Пример задачи: Дано k отсортированных списков с общим количеством n элементов. Надо их соединить в один отсортированный список за время O(nlogk), используя пирамиду.

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 экзамену.

Перечень вопросов к экзамену (КИМ №1)

1. Программные системы. Основные проблемы разработки сложных программных систем. Принципы работы со сложными системами
2. Жизненный цикл ПО. Стандартизация жизненного цикла в системе государственных стандартов ЕСПД и ГОСТ Р ИСО/МЭК 12207.
3. Модели жизненного цикла (каскадная, спиральная, формальные и др.).
4. Гибкие методологии разработки (XP, Scrum, RUP и др.).
5. Анализ предметной области и требования к ПО. Функциональные и нефункциональные требования. Разработка требований. Спецификация требований к программному обеспечению, техническое задание.
6. Архитектурное и детальное проектирование. Основные принципы проектирования (структурирование системы, моделирование управления, модульная декомпозиция).
7. Архитектура распределенных систем. Характеристики распределенных систем. Архитектура клиент-сервер (модели тонкого и толстого клиента). Архитектура распределенных объектов.
8. Объектно-ориентированное проектирование. SOLID. Процесс объектно-ориентированного проектирования: окружение системы и модели ее использования, проектирование архитектуры, определение объектов, модели архитектуры, специфицирование интерфейсов объектов.
9. Проектирование с повторным использованием компонентов. Достоинства и недостатки. Покомпонентная разработка. Семейства приложений. Проектные паттерны (фасад, одиночка, адаптер, фабрика, команда, стратегия, шаблонный метод, наблюдатель, декоратор, итератор).
10. Проектирование интерфейса пользователя. Принципы проектирования интерфейсов пользователя. Взаимодействие с пользователем. Представление информации. Средства поддержки пользователя. Оценивание интерфейса.
11. Качество ПО и методы его контроля. Качество программного обеспечения. Методы контроля качества. Тестирование. Проверка на моделях. Виды и уровни тестирования. Стратегии восходящего и нисходящего тестирования. Методы «белого» и «черного» ящика. Автоматизированное и ручное тестирование. Разработка через тестирование (TDD). Покрывание кода тестами.
12. Управление проектом. Инструментальные средства управления проектами. Управление персоналом. Права и обязанности членов коллектива. Мотивация. Организация совместной работы коллектива.
13. Структуры данных.
14. Массив, стек, очередь, список. Словарь. Реализация, методы работы с этими структурами данных.
15. Бинарные деревья. Применение деревьев для поиска.
16. Самобалансирующиеся деревья. AVL-деревья.

17. Б-деревья, их применение для внешней сортировки.
18. Список с пропусками.
19. Пирамида (двоичная куча). Ее реализация на основе массива.
20. Хеш-таблицы. Разрешение коллизий методом цепочек. Разрешение коллизий с помощью открытой адресации: применение линейного, квадратичного метода и метода двойного хеширования.
21. Алгоритмы. Модель вычислений для сравнения алгоритмов. Наихудшее и среднее время работы. Обозначения O и Θ .
22. Сортировки.
23. Алгоритмы внутренней сортировки: сортировки сравнениями: вставками, выбором и обменами. Оценка сложности работы алгоритмов внутренней сортировки.
24. Пирамидальная сортировка. Время ее работы.
25. Быстрая сортировка. Время ее работы.
26. Основная теорема для оценки работы рекурсивных алгоритмов (без доказательства).
27. Алгоритмы поиска подстрок.
28. Алгоритм простого поиска подстрок.
29. Алгоритм Рабина-Карпа.
30. Алгоритм Кнута-Морриса-Пратта.
31. Алгоритм Бойера- Мура.
32. Алгоритмы на графах.
33. Способы задания графов.
34. Алгоритмы обхода графов в ширину и глубину.
35. Алгоритм нахождения двусвязных компонент графа.
36. Остовные деревья минимальной стоимости и алгоритмы их построения.
37. Алгоритмы нахождения кратчайших расстояний между вершинами (Дейкстра, Флойд, Беллмана-Форда).
38. Задача о максимальном потоке (алгоритм Эдмонса-Карпа).

Критерии оценки ответов на вопросы зачета с оценкой

Для оценивания результатов обучения на зачете с оценкой 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

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

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

ОПК-4.1. Способен организовывать защиту информации в компьютерных системах и сетях (по областям применения).

ОПК-4.3. Способен разрабатывать и анализировать корректность политики информационной безопасности компьютерных систем и сетей (по областям применения).

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

Вопрос 1. К основным преимуществам автоматизированных тестов можно отнести

1. Оперативное информирование о состоянии качества продукта
2. Удобство их запуска из консоли
3. Исключение человеческого фактора при проверке
4. Гибкость составления тестовых наборов

Вопрос 2. В качестве дополнительной информации к сообщению об ошибке можно приложить

- a) Снимок экрана с дефектом
- b) Логи во время возникновения дефекта
- c) Видео воспроизведения дефекта
- d) Тестовые данные необходимые для воспроизведения дефекта

Вопрос 3. Тестирование методом белого, серого и черного ящиков - это тестирование, которое классифицировано

- a) По знанию тестируемого объекта
- b) По степени подготовленности к тестированию
- c) По моменту проведения
- d) По характеру сценариев

Вопрос 4. Обеспечения качества - это процесс или результат формирования и поддержки требуемых характеристик и свойств продукции на этапах:

- a) создания
- b) хранения
- c) транспортирования
- d) эксплуатации

Вопрос 5. Что из перечисленного относится к внутренним процедурам обеспечения качества?

- a) тестирование спецификации
- b) интеграционное тестирование
- c) юнит-тесты
- d) обработка заявок в техническую поддержку

Вопрос 6. Позволяет экономить время на генерации тестовых данных

- a) Нагрузочное тестирование
- b) Объемное тестирование
- c) Автоматизированное тестирование
- d) Фокус-тестирование

Вопрос 7. В описание ошибки необходимо указывать

- a) Последовательность действий для воспроизведения
- b) Ожидаемый результат
- c) ФИО потенциально виновного разработчика
- d) Фактический результат

Вопрос 8. Тестирование на уровне кода, когда проверяется работа конкретной функции или метода- это

- a) Модульное тестирование
- b) Тестирование методом свободного поиска

с) Функциональное тестирование

д) Интеграционное тестирование

Вопрос 9. Тестирование спецификации включает проверки на:

а) грамотность, политкорректность

б) логику, полноту

с) функциональность, однозначность

д) понятность, удобство

Вопрос 10. Что из перечисленного НЕ входит в основные этапы обеспечения качества?

а) оценка уровня качества имеющихся на рынке аналогичных изделий, анализ требований покупателей

б) пооперационный контроль в процессе производства

с) исправление критических дефектов

д) контроль качества изделия в условиях эксплуатации (после продажи)

1	2	3	4	5	6	7	8	9	10
а,с	а,б,с,д	а	а,б,с,д	а, с	с	а,б,д	а	б,с	с

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