

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сахалинский государственный университет»

Кафедра информатики

УТВЕРЖДЕН
на заседании кафедры
«19» марта 2024 г., протокол № 8
Исполняющий обязанности
заведующего кафедрой



Осипов Г.С.

**ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

Б1.О.17 Объектно-ориентированное программирование

Направление подготовки
10.03.01 Информационная безопасность
профиль

Безопасность автоматизированных систем (по отрасли или в сфере профессиональной деятельности)

**Уровень высшего образования
БАКАЛАВРИАТ**

Южно-Сахалинск
2024 г.

1. Формируемые компетенции и индикаторы их достижения по дисциплине (модулю)

Код компетенции	Содержание компетенции	Код и наименование индикатора достижения компетенции
ОПК-7	Способен использовать языки программирования и технологии разработки программных средств для решения задач профессиональной деятельности;	ОПК-7.1 - Знает основные технологии разработки программных средств для решения задач в области профессиональной деятельности; ОПК-7.2 - Умеет применять языки программирования для решения профессиональных задач; ОПК-7.3 - Владеет навыками выбора и разработки алгоритмов при решении типовых задач программирования, а также навыками разработки и тестирования программ по поставленной спецификации.

2. Паспорт фонда оценочных средств по дисциплине (модулю)

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
4 семестр			
1.	Тема 1. Перечисления и структуры. Перечисления и структуры как пользовательские типы значений. Базовый класс перечислений.	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
2.	Тема 2. Исключения. Системные исключения и их обработка. Свойства исключений. Управление программой с помощью исключений. Исключения в арифметических выражениях. Генерация исключений. Пользовательские классы исключений.	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
3.	Тема 3. Делегаты и события. Синтаксис делегатов. Массивы делегатов. Многоадресные (групповые) экземпляры делегатов. Делегаты и обратные вызовы. Анонимные методы. Лямбда-выражения. События.	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
4.	Тема 4. Обобщения. Обобщения как средство абстракции. Декларации обобщённых классов. Ограничения типизирующих параметров. Обобщённые структуры. Обобщённые интерфейсы. Обобщённые методы. Обобщённые делегаты.	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
5.	коллоквиумы	ОПК-7	контрольные вопросы, вопросы к коллоквиуму
6.	экзамен	ОПК-7	контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
5 семестр			

7.	Тема 1. Организация системы ввода- вывода в C# на потоках	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
8.	Тема 2. Пространства имен, препроцессор, сборки	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
9.	Тема 3. Динамическая идентификация типов, рефлексия и атрибуты.	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
10.	Тема 4. Небезопасный код, указатели, обнуляемые типы.	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
11.	Тема 5. Основы LINQ	ОПК-7	Задания к лабораторным работам, контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену
12.	Тема 6. Коллекции, перечислители и итераторы.	ОПК-7	контрольные вопросы, вопросы к коллоквиуму
13.	экзамен	ОПК-7	контрольные вопросы, вопросы к коллоквиуму, вопросы к экзамену

4 семестр

Лабораторное занятие №1 (8 ч.)

Тема Перечисления и структуры

Вопросы для обсуждения:

1. Перечисления. Базовый класс перечислений.
2. Структуры как пользовательские типы значений.
3. Особенности реализации перечислений и структур.

Примерные задания

1. Индивидуальные задания по теме «Перечисления» (20 вариантов)

Пример. В задаче Разрабатываемый класс должен содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы, свойства, индексы, перечисления. Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса. В программе должна выполняться проверка всех разработанных элементов класса.

Задача об инфекции стригущего лишая (Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1981)

Промоделировать процесс распространения инфекции – стригущего лишая по участку кожи размером $p \times p$ (p – нечетное) клеток. Предполагается, что исходной зараженной клеткой кожи является центральная. В каждый интервал времени пораженная инфекцией клетка может с вероятностью 0,5 заразить любую из соседних здоровых клеток. По прошествии 6 единиц времени зараженная клетка становится невосприимчивой к инфекции. Возникший иммунитет действует в течение последующих 4 единиц времени, а затем клетка оказывается здоровой.

В ходе моделирования описанного процесса необходимо выдавать текущее состояние исследуемого участка кожи в каждый интервал времени, отмечая зараженные, невосприимчивые к инфекции и здоровые клетки.

2. Индивидуальные задания по теме «Структуры» (20 вариантов)

Пример.

Вариант 1

Описать структуру с именем STUDENT, содержащую следующие поля:

- фамилия и инициалы;
- номер группы;
- успеваемость (массив из пяти элементов).

Написать программу, выполняющую следующие действия:

- ввод с клавиатуры данных в массив, состоящий из десяти структур типа STUDENT (записи должны быть упорядочены по возрастанию номера группы);

вывод на экран фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,0 (если таких студентов нет, вывести соответствующее сообщение).

Лабораторное занятие №2 (8 ч.)

Тема Исключения

Вопросы для обсуждения:

1. Системные исключения и их обработка.
2. Свойства исключений.
3. Управление программой с помощью исключений.
4. Исключения в арифметических выражениях.
5. Генерация исключений.
6. Пользовательские классы исключений.
7. Особенности реализации.

Примерные задания

1. Индивидуальные задания по теме «Исключения» (20 вариантов).

Пример. Разрабатываемый класс должен содержать следующие элементы: скрытые поля, конструкторы с параметрами и без параметров, методы, свойства. Методы и свойства должны обеспечивать непротиворечивый, полный, минимальный и удобный интерфейс класса. При возникновении ошибок должны выбрасываться исключения. В программе должна выполняться проверка всех разработанных элементов класса.

Вариант 1

Описать класс, реализующий десятичный счетчик, который может увеличивать или уменьшать свое значение на единицу в заданном диапазоне. Предусмотреть инициализацию счетчика значениями по умолчанию и произвольными значениями. Счетчик имеет два метода: увеличения и уменьшения, и свойство, позволяющее получить его текущее состояние. При выходе за границы диапазона выбрасываются исключения. Написать программу, демонстрирующую все разработанные элементы класса.

Крестики-нолики

Написать программу, позволяющую играть на бесконечном поле в "крестики-нолики":

- а) игроку с компьютером;
- б) двум игрокам.

Если в качестве игрока выступает компьютер, программа делает первый ход. Делая очередной ход, проанализировать ситуацию, рассчитывая ходы противника вперед на 1-2 хода, и в результате проведенного анализа, выбрать оптимальный результат.

Лабораторное занятие №3 (10 ч.)

Тема Делегаты и события

Вопросы для обсуждения:

1. Синтаксис делегатов.
2. Массивы делегатов.
3. Многоадресные (групповые) экземпляры делегатов.
4. Делегаты и обратные вызовы.
5. Анонимные методы.
6. Лямбда-выражения.
7. События
8. Особенности реализации.

Примерные задания

1. Разработать программу, использующую делегаты.

1. Программа должна быть разработана в виде консольного приложения на языке C#.

2. Определить делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа.

3. Написать метод, соответствующий данному делегату.

4. Написать метод, принимающий разработанный Вами делегат, в качестве одного из входных параметров. Осуществить вызов метода, передавая в качестве параметра-делегата:

- ☐ метод, разработанный в пункте 3;
- ☐ анонимный метод.

2. Лабораторная работа. Лямбда-выражения

Задачи на использование одиночных и блочных лямбда-выражений. Описать делегат с требуемой сигатурой. Используя одиночное или блочное лямбда-выражение реализовать основной алгоритм задачи. Осуществить использование делегата в программе с применением введенных пользователем исходных данных.

Задание 1

Продemonстрировать работу лямбда-выражения, которое получает входным параметром целое число x и вычисляет количество цифр «5» в этом числе.

3. Индивидуальные задания по теме «Делегаты» (9 заданий)

Пример.

Задание 6.

Использование анонимных методов.

Продemonстрировать вызов анонимного метода, который получает входные параметры и возвращает значение. Для этого реализовать функцию, которая находит площадь треугольника по формуле Герона. Функция получает входным параметром значения трех сторон треугольника a , b , c . Если введены некорректные значения a , b , c , то функция возвращает -1 (отрицательное число).

4. Лабораторная работа. События

Пример.

Задание 4.

Использование обобщенного делегата `EventHandler<EventArgs>`

Обобщенный делегат `EventHandler<EventArgs>` реализован в среде .NET. Этот делегат предназначен для замены собственного делегата события.

На примере задачи 3:

1. продemonстрировать применение обобщенного делегата `EventHandler`.
2. использовать объект-заполнитель `Empty` из класса `EventArgs`.

5. Крестики-нолики

Написать программу, позволяющую играть на бесконечном поле в "крестики-нолики":

- а) игроку с компьютером;
- б) двум игрокам.

Если в качестве игрока выступает компьютер, программа делает первый ход. Делая очередной ход, проанализировать ситуацию, рассчитывая ходы противника вперед на 1-2 хода, и в результате проведенного анализа, выбрать оптимальный результат.

Лабораторное занятие №4 (10 ч.)

Тема **Обобщения в C#.**

Вопросы для обсуждения:

1. Обобщения как средство абстракции.
2. Декларации обобщённых классов.
3. Ограничения типизирующих параметров.
4. Обобщённые структуры.
5. Обобщённые интерфейсы.
6. Обобщённые методы.
7. Обобщённые делегаты.
8. Особенности реализации.

Пример.

1. Разработать программу, использующую обобщенные делегаты.
 1. Программа должна быть разработана в виде консольного приложения на языке C#.
 2. Определить обобщенный делегат, принимающий несколько параметров различных типов и возвращающий значение произвольного типа.
 3. Написать метод, соответствующий данному делегату.
 4. Написать метод, принимающий обобщенный делегат `Func<>` или `Action<>`, соответствующий сигнатуре разработанного Вами делегата, в качестве одного из входных параметров. Осуществить вызов метода, передавая в качестве параметра-делегата:
 - ☐ метод, разработанный в пункте 3;
 - ☐ лямбда-выражения.
2. Применение библиотечных обобщенных типов делегатов `Func<>` и `Action<>`. Необходимо написать метод для формирования массива из заданного количества значений членов ряда Фибоначчи и обобщенный метод для вывода в консольное окно значений элементов одномерного массива. В основной программе объявить два делегата для представления указанных методов и, используя экземпляры делегатов, вывести значения первых семи членов ряда Фибоначчи
3. *Итальянская игра "Математико".*

Имеется набор из 52 карточек, на которых записаны числа от 1 до 13, при этом каждая карточка встречается 4 раза. Требуется написать программу, которая позволяет имитировать игру человека с компьютером. Имеется поле с 25 клетками. Программа случайным образом извлекает одну из имеющихся карточек и выдает записанное на ней число. Каждый игрок заносит это число в одну из клеток квадрата. Так продолжается до тех пор, пока не будут заполнены все клетки квадрата.

1	1	7	1	7	(80)
2	10	2	13	2	(40)
5	12	13	5	7	(10)
3	3	3	11	3	(160)
4	12	4	13	12	(20)
(20)	(50)	(10)	(10)	(10)	(160)

По окончании игры заполнение соответствующего квадрата оценивается определенным количеством очков. Цель игры – разместить числа в клетках таблицы так, чтобы набрать наибольшее количество очков в соответствии с таблицей:

комбинации чисел	в ряду или столбце	по диагонали
за 2 одинаковых числа	10 очков	20 очков
за 2 пары одинаковых чисел	20 очков	30 очков
за 3 одинаковых числа	40 очков	50 очков
за 3 одинаковых числа и два других одинаковых числа	80 очков	90 очков
за 4 одинаковых числа	160 очков	170 очков
за 5 последовательных чисел, но не обязательно расположенных по порядку	50 очков	60 очков
за три раза по 1 и по два раза по 13	100 очков	110 очков
за числа 1, 13, 12, 11 и 10, но не обязательно расположенных по порядку	150 очков	160 очков
за четыре единицы	200 очков	210 очков

Необходимо написать оптимальную стратегию заполнения квадрата.

5 семестр

Лабораторное занятие №1 (8 ч.)

Тема Организация системы ввода- вывода в C# на потоках

Вопросы для обсуждения:

1. Байтовые и символьные потоки.
2. Классы потоков. Классы-оболочки.
3. Консольный ввод-вывод.
4. Класс FileStream и байтовый ввод-вывод в файл.
5. Символьный ввод-вывод в файл.
6. Класс BinaryReader.
7. Файлы с произвольным доступом.
8. Применение класса MemoryStream.
9. Классы File и FileInfo.

1. Индивидуальные задания по теме Фалы №1 (12 заданий)

Пример.

Средства System.IO.File для работы с текстовыми файлами

1. Дан текстовый файл. Определить количество пустых строк в этом файле, используя статический метод ReadLines класса System.IO.File.
2. Дан csv-файл, содержащий целые числа. Найти сумму чисел в каждой строке файла (решение оформить в виде функции, возвращающей массив целых чисел). Для пустой строки следует возвращать ноль.

Замечание 1. Каждая строка файла в формате CSV (comma-separated values — значения, разделённые запятыми) содержит значения, разделённые запятыми.

Указание 1. Использовать метод Split для извлечения чисел из строк и метод int.Parse для преобразования их к типу integer для суммирования.

Замечание 2. Задачу можно решать с использованием явных циклов, либо с использованием методов последовательностей.

Лабораторное занятие №2 (4 ч.)

Тема Пространства имен, препроцессор, сборки.

Вопросы для обсуждения:

1. Пространства имен.
2. Препроцессор.
3. Сборки и модификатор доступа internal.
4. Особенности реализации.

Пример.

1. Задачи на использование директив препроцессора.
2. Волчий остров (Ван Тассел Д. Стил, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1981)

Волчий остров размером 20 x 20 заселен дикими кроликами, волками и волчицами. Имеется по несколько представителей каждого вида. Кролики довольно глупы: в каждый момент времени они с одинаковой вероятностью 1/9 передвигаются в один из восьми соседних квадратов (за исключением участков, ограниченных береговой линией) или просто сидят неподвижно. Каждый кролик с вероятностью 0,2 превращается в двух кроликов. Каждая волчица передвигается случайным образом, пока в одном из соседних восьми квадратов не окажется кролик, за которым она охотится. Если волчица и кролик оказываются в одном квадрате, волчица съедает кролика и получает одно очко. В противном случае она теряет 0,1 очка. Волки и волчицы с нулевым количеством очков умирают.

В начальный момент времени все волки и волчицы имеют 1 очко. Волк ведет себя подобно волчице до тех пор, пока в соседних квадратах не исчезнут все кролики; тогда, если волчица находится в одном из восьми близлежащих квадратов, волк гонится за ней. Если волк и волчица окажутся в одном квадрате, и там нет кролика, которого нужно съесть, они производят потомство случайного пола.

Написать программу, реализующую данную экологическую модель и наблюдать за изменением популяции в течение некоторого периода времени

Лабораторное занятие №3 (6 ч.)

Тема **Динамическая идентификация типов, рефлексия и атрибуты**

Вопросы для обсуждения:

1. Динамическая идентификация типов.
2. Рефлексия. Применение рефлексии.
3. Основы применения атрибутов: создание, присоединение, получение атрибутов.
4. Встроенные атрибуты.
5. Особенности реализации.

Примерное задание.

С помощью рефлексии проверить текущую версию сборки, сконструировать объект искомого класса, а затем вызвать методы, определенные в этом классе. Затем скомпилируйте этот файл в библиотеку DLL под именем MyClass.dll. Составьте программу, в которой применяется библиотека MyClass.dll

Лабораторное занятие №4 (6 ч.)

Тема **Небезопасный код, указатели, обнуляемые типы**

Вопросы для обсуждения:

1. Обнуляемые типы.
2. Частичные типы.
3. Частичные методы.
4. Создание объектов динамического типа.
5. Возможность взаимодействия с моделью COM.
6. Особенности реализации.

Примерное задание.

1. Используя стек, решить следующую задачу:

В текстовом файле LOG записано без ошибок логическое выражение (ЛВ) в следующей форме:

`<ЛВ>::=true | false | (□<ЛВ>) | (<ЛВ> □ <ЛВ>) | (<ЛВ> □ <ЛВ>)`

Вычислить (как Boolean) значение этого выражения. Знаки □, □, □ означают соответственно отрицание, конъюнкцию и дизъюнкцию.

2. Дан некоторый текст произвольной длины. Необходимо проверить его симметричность, используя динамический список.
3. Морской бой.

Написать программу игры в морской бой игрока с компьютером. Программа должна позволять расставлять корабли на поле размером 10 x 10, контролировать правильность их расстановки, давать противникам возможность поочередно делать ходы и выдавать соответствующие информационные сообщения. Во время хода компьютера, необходимо анализировать предыдущие ходы, и следующий ход делать на основе проведенного анализа

Лабораторное занятие №5 (6 ч.)

Тема Основы LINQ

Вопросы для обсуждения:

1. Основы LINQ.
2. Отбор запрашиваемых значений.
3. Группирование результатов запросов.
4. Анонимные типы.
5. Режимы выполнения запросов: отложенный и немедленный.
6. Деревья выражений.
7. Особенности реализации.

Примерное задание. 1. Используя запрос LINQ, получить положительные значения, содержащиеся в массиве целых чисел. Содержимое массива изменить в промежутке между двумя последовательно производимыми попытками выполнить один и тот же запрос. Соблюдить соответствие типов переменной запроса и переменной диапазона. Из массива вывести только те значения, которые удовлетворяют условию. Извлечь значения из массива по возрастанию.

Примерное задание. 2. Определяется массив символьных строк. В ряде этих строк содержатся адреса Интернета. По запросу извлечь те строки, которые содержат более четырех символов и оканчиваются на ".net".

Лабораторное занятие №6 (6 ч.)

Тема Коллекции, перечислители и итераторы.

Вопросы для обсуждения:

1. Обзор коллекций.
2. Необобщенные коллекции.
3. Хранение отдельных битов в классе коллекции BitArray.
4. Специальные коллекции.
5. Обобщенные коллекции.
6. Параллельные коллекции.

Пример.

Разработать программу, реализующую работу с коллекциями.

1. Программа должна быть разработана в виде консольного приложения на языке C#.
2. Создать объекты классов «Прямоугольник», «Квадрат», «Круг».
3. Для реализации возможности сортировки геометрических фигур для класса «Геометрическая фигура» добавить реализацию интерфейса IComparable. Сортировка производится по площади фигуры.
4. Создать коллекцию класса ArrayList. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
5. Создать коллекцию класса List<Figure>. Сохранить объекты в коллекцию. Отсортировать коллекцию. Вывести в цикле содержимое коллекции.
6. Модифицировать класс разреженной матрицы Matrix (представлен в разделе «Вспомогательные материалы для выполнения лабораторных работ») для работы с тремя измерениями – x,y,z. Вывод элементов в методе ToString() осуществлять в том виде, который Вы считаете наиболее удобным. Разработать пример использования разреженной матрицы для геометрических фигур.
7. Реализовать класс «SimpleStack» на основе односвязного списка. Класс SimpleStack наследуется от класса SimpleList (из «Вспомогательные материалы для выполнения лабораторных работ»).

Необходимо добавить в класс методы:

- ☐ public void Push(T element) – добавление в стек;
- ☐ public T Pop() – чтение с удалением из стека.

8. Пример работы класса SimpleStack реализовать на основе геометрических фигур.

Форма контроля	За одну работу		Всего	
	Мин. баллов	Макс. баллов	Мин. баллов	Макс. баллов
Текущий контроль:				
Активная работа на занятии	0,25	0,5	9	18
Выполнение домашнего задания	0,75	0,75	27	27

Выполнение заданий самостоятельной работы	1	3	1	3
<i>коллоквиум</i>	1	3	3	9
Промежуточная аттестация (экзамен)			20	43
Итого за семестр			60	100

Примерные вопросы к экзамену (4 семестр)

1. Перечисления и структуры как пользовательские типы значений. Базовый класс перечислений.
2. Перечисления и структуры как пользовательские типы значений. Структуры.
3. Системные исключения и их обработка. Свойства исключений.
4. Управление программой с помощью исключений.
5. Исключения в арифметических выражениях. Генерация исключений. Пользовательские классы исключений.
6. Делегаты и события. Синтаксис делегатов. Массивы делегатов.
7. Делегаты и события. Многоадресные (групповые) экземпляры делегатов. Делегаты и обратные вызовы. Анонимные методы. Лямбда-выражения.
8. Делегаты и события. События.
9. Обобщения как средство абстракции. Декларации обобщённых классов. Ограничения типизирующих параметров.
10. Обобщения как средство абстракции. Обобщённые структуры.
11. Обобщённые интерфейсы. Обобщённые методы.
12. Обобщённые делегаты.

Примерные вопросы к экзамену (5 семестр)

1. Организация системы ввода- вывода в C# на потоках. Байтовые и символьные потоки. Классы потоков. Классы-оболочки.
2. Организация системы ввода- вывода в C# на потоках. Консольный ввод-вывод. Класс FileStream и байтовый ввод-вывод в файл. Символьный ввод-вывод в файл. Класс BinaryReader.
3. Файлы с произвольным доступом. Применение класса MemoryStream. Классы File и FileInfo.
4. Пространства имен, препроцессор, сборки
5. Динамическая идентификация типов.
6. Рефлексия и ее применение. Атрибуты. Встроенные атрибуты.
7. Небезопасный код
8. Обнуляемые типы. Частичные типы, частичные методы.
9. Создание объектов динамического типа. Возможность взаимодействия с моделью COM.
10. Дружественные сборки.
11. Основы LINQ. Отбор запрашиваемых значений. Сортировка результатов запроса, их группирование. Анонимные типы. Создание группового запроса.
12. Деревья выражений. Методы расширения.
13. Коллекции. Необобщенные, специальные, обобщенные, параллельные коллекции. Сохранение объектов, определяемых пользователем классов, в коллекции.
14. Реализация и применение интерфейса IComparable.
15. Доступ к коллекции с помощью перечислителя. Реализация интерфейсов IEnumerable и IEnumerator.
16. Применение итераторов. Прерывание итераторов. Создание именованного и обобщенного итератора.

Критерии оценки:

— **оценка «отлично»** выставляется студенту, если студент свободно ориентируется в теоретическом материале; умеет изложить и корректно оценить различные подходы к излагаемому материалу, способен сформулировать и доказать собственную точку зрения; обнаруживает свободное владение понятийным аппаратом; демонстрирует готовность применять теоретические знания в практической деятельности и полное освоение показателей формируемых компетенций;

— **оценка «хорошо»** выставляется студенту, если студент хорошо ориентируется

в теоретическом материале; имеет представление об основных подходах к излагаемому материалу; знает определения основных теоретических понятий излагаемой темы, в основном демонстрирует готовность применять теоретические знания в практической деятельности и освоение большинства показателей формируемых компетенций;

– **оценка «удовлетворительно»** выставляется студенту, если студент может ориентироваться в теоретическом материале; в целом имеет представление об основных понятиях излагаемой темы, частично демонстрирует готовность применять теоретические знания в практической деятельности и освоение некоторых показателей формируемых компетенций;

– **оценка «неудовлетворительно»** выставляется студенту, если студент не ориентируется в теоретическом материале; не сформировано представление об основных понятиях излагаемой темы, не демонстрирует готовность применять теоретические знания в практической деятельности и освоение показателей формируемых компетенций.

Составитель *Н. Ваши*

Вашакидзе Н.С.

«12» марта 2024 г.