Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Чижов Александр Петрович
Должность: Директор филиала СибГУ в г. Лесосибирске
Дата подписания: 13.07.2023 10:14:14
Уникальный программный ключ:
bdf6e99bfcc4944b52cae00e83cf259c6c8b1a3963a47694-36e0cbf0e61

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Лесосибирский филиал федерального государственного бюджетного
образовательного учреждения высшего образования
«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

Направление подготовки
09.03.01 Информатика и вычислительная техника

Направленность (профиль) образовательной программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования
бакалавриат

Форма обучения
очная, заочная

Красноярск 2022

Рабочая программа дисциплины составлена в соответствии с актуализированным федеральным государственным образовательным стандартом высшего образования – бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденным приказом Министерства науки и высшего образования Российской Федерации 19.09.2017г. №929

Разработчики рабочей программы дисциплины:

к.т.н., доцент, заведующий кафедрой информационных и технических систем	_____	П.А. Егармин
должность, учёная степень, учёное звание	подпись	И.О. Фамилия
Руководитель ОПОП, к.т.н., доцент, заведующий кафедрой информационных и технических систем	_____	П.А. Егармин
должность, учёная степень, учёное звание	подпись	И.О. Фамилия

Рабочая программа дисциплины рассмотрена на заседании кафедры информационных и технических систем от «11» октября 2022г. протокол № 10а

Заведующий кафедрой, к.т.н., доцент	_____	П.А. Егармин
должность, учёная степень, учёное звание	подпись	И.О. Фамилия

Рабочая программа дисциплины рассмотрена на заседании научно-методического совета филиала от «11» октября 2022г., протокол № 4

Председатель НМС филиала, к.т.н., доцент	_____	С.В. Соболев
должность, учёная степень, учёное звание	подпись	И.О. Фамилия

Рабочая программа дисциплины утверждена в составе ОПОП решением Ученого совета СибГУ им. М.Ф. Решетнева №5 от 28.10.2022г.

АННОТАЦИЯ
Рабочей программы дисциплины
Программирование

(наименование дисциплины)

Направление подготовки (Специальность)	09.03.01 Информатика и вычислительная техника
Направленность (профиль)	Программное обеспечение средств вычислительной техники и автоматизированных систем

Объем дисциплины составляет 18 зачетных единиц, 648 часов.

Цель и задачи изучения дисциплины

Цель изучения дисциплины	– изучение теоретических основ и приобретение практических навыков технологии программирования, базирующихся на фундаментальных принципах построения программного продукта и объектно-ориентированного программирования
Задачи изучения дисциплины:	– развитие алгоритмического мышления, необходимого для последующего изучения смежных дисциплин; – развитие способности разрабатывать алгоритмы и программы, пригодные для практического применения; – развитие способности к интеграции программных модулей и компонент, и верификации выпусков программного продукта

Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенций

Код компетенции	Содержание компетенции	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенции
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	**ОПК-8.1.** Применяет методы проектирования программного обеспечения **ОПК-8.2.** Использует современные языки программирования для разработки алгоритмов и программ **ОПК-8.3.** Владеет методами отладки и тестирования программ	**Знать:** – основы алгоритмизации и программирования; – современные среды разработки программного обеспечения; – возможности современных и перспективных средств разработки программных продуктов; – синтаксис и основные особенности одного из современных языков программирования. **Уметь:** – составлять алгоритмы; – писать и отлаживать коды на языке программирования; – тестировать работоспособность программы; – интегрировать программные модули. **Владеть:** – языком программирования; – методами отладки и тестирования работоспособности программы

Место дисциплины в структуре ОПОП

Дисциплина «Программирование» (Б1.О.12) входит в обязательную часть блока Б1 «Дисциплины (модули)».

Краткое содержание дисциплины

Раздел 1. Основы алгоритмизации и программирования.

Раздел 2. Объектно-ориентированное программирование.

Раздел 3. Расширенные возможности.

Форма промежуточной аттестации

Экзамен, курсовая работа.

Оглавление

1. Цель и задачи изучения дисциплины 2
2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенций 2
3. Место дисциплины в структуре ОПОП 2
4. Объем дисциплины и виды учебной работы 3
5. Содержание дисциплины 4
5.1. Разделы дисциплины и виды занятий 4
5.2. Занятия лекционного типа 6
5.3. Занятия семинарского типа 7
5.4. Занятия в форме практической подготовки 9
6. Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине 9
7. Учебно-методическое обеспечение дисциплины 9
7.1. Рекомендуемая литература 9
7.2. Перечень современных профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины 10
7.3. Методические указания для обучающихся по освоению дисциплины 10
8. Материально-техническое обеспечение дисциплины 11

1. Цель и задачи изучения дисциплины

Цель изучения дисциплины	– изучение теоретических основ и приобретение практических навыков технологии программирования, базирующихся на фундаментальных принципах построения программного продукта и объектно-ориентированного программирования
Задачи изучения дисциплины:	– развитие алгоритмического мышления, необходимого для последующего изучения смежных дисциплин; – развитие способности разрабатывать алгоритмы и программы, пригодные для практического применения; – развитие способности к интеграции программных модулей и компонент, и верификации выпусков программного продукта

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенций

Код компетенции	Содержание компетенции	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенции
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	**ОПК-8.1.** Применяет методы проектирования программного обеспечения **ОПК-8.2.** Использует современные языки программирования для разработки алгоритмов и программ **ОПК-8.3.** Владеет методами отладки и тестирования программ	**Знать:** – основы алгоритмизации и программирования; – современные среды разработки программного обеспечения; – возможности современных и перспективных средств разработки программных продуктов; – синтаксис и основные особенности одного из современных языков программирования. **Уметь:** – составлять алгоритмы; – писать и отлаживать коды на языке программирования; – тестировать работоспособность программы; – интегрировать программные модули. **Владеть:** – языком программирования; – методами отладки и тестирования работоспособности программы

3. Место дисциплины в структуре ОПОП

Дисциплина «Программирование» (Б1.В.02) входит в обязательную часть блока Б1 «Дисциплины (модули)».

Изучение курса связано с дисциплинами: «Информатика», «Математика», «Вычислительная математика», «Математическая логика и теория алгоритмов».

Знания, умения и навыки, полученные в ходе изучения дисциплины, являются необходимыми для изучения дисциплин: «Объектно-ориентированное программирование и проектирование», «Современные языки программирования», «Программирование под

Интернет», «Технология разработки программного обеспечения», «Тестирование и отладка программного обеспечения», «Разработка мобильных приложений», а также для написания выпускной квалификационной работы.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 18 зачетных единиц, 648 часов.

а) очная форма

Вид учебной работы / номер семестра в УП	Всего, зачетных единиц (акад. часов)	Семестр	Семестр	Семестр
Номер семестра		2	3	4
Общая трудоемкость дисциплины	**18 (648)**	**6 (216)**	**6 (216)**	**6 (216)**
Контактная работа с преподавателем:	**9 (324)**	**3 (108)**	**3 (108)**	**3 (108)**
занятия лекционного типа	3 (108)	1 (36)	1 (36)	1 (36)
занятия семинарского типа	6 (216)	2 (72)	2 (72)	2 (72)
в том числе: семинары				
практические занятия				
практикумы				
лабораторные работы	6 (216)	2 (72)	2 (72)	2 (72)
коллоквиумы				
иные аналогичные занятия				
в том числе: курсовое проектирование				
групповые консультации				
индивидуальная работа с преподавателем				
иная контактная внеаудиторная работа				
Самостоятельная работа обучающихся:	**9 (324)**	**3 (108)**	**3 (108)**	**3 (108)**
изучение теоретического курса (ТО)	7 (252)	3 (108)	3 (108)	1 (36)
индивидуальные задания (ИЗ)				
расчетно-графические работы (РГР)				
реферат, эссе (Р)				
курсовое проектирование (КР/КП)	2 (72)			2 (72)
контрольные работы (Кн.р)				
другие виды самостоятельной работы				
Форма промежуточной аттестации (зачет, зачет с оценкой, экзамен, курсовой проект, курсовая работа)		**Экзамен**	**Экзамен**	**Экзамен, курсовая работа**

б) заочная форма

Вид учебной работы / номер семестра в УП	Всего, зачетных единиц (акад. часов)	Семестр	Семестр	Семестр	Семестр
Номер семестра		2	3	4	5
Общая трудоемкость дисциплины	**18 (648)**	**1 (36)**	**6 (216)**	**6 (216)**	**5 (180)**
Контактная работа с преподавателем:	**2,33 (84)**	**0,06(2)**	**0,72 (26)**	**0,78 (28)**	**0,78 (28)**
занятия лекционного типа	0,83 (30)	0,06(2)	0,22 (8)	0,28 (10)	0,28 (10)
занятия семинарского типа	1,5 (54)		0,5 (18)	0,5 (18)	0,5 (18)
в том числе: семинары					
практические занятия					
практикумы					
лабораторные работы	1,5 (54)		0,5 (18)	0,5 (18)	0,5 (18)
коллоквиумы					
иные аналогичные занятия					
в том числе: курсовое проектирование					
групповые консультации					
индивидуальная работа с преподавателем					
иная контактная внеаудиторная работа					
Самостоятельная работа обучающихся:	**15,67 (564)**	**0,94(34)**	**5,28 (190)**	**5,22 (188)**	**4,22 (152)**

изучение теоретического курса (ТО)	12,66 (456)	0,94 (34)	5,28 (190)	5,22 (188)	1,22 (44)
индивидуальные задания (ИЗ)					
расчетно-графические работы (РГР)					
реферат, эссе (Р)					
курсовое проектирование (КР/КП)	3 (108)				3 (108)
контрольные работы (Кн.р)					
другие виды самостоятельной работы					
Форма промежуточной аттестации (зачет, зачет с оценкой, экзамен, курсовой проект, курсовая работа)			**Экзамен**	**Экзамен**	**Экзамен, курсовая работа**

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

а) очная форма

№ п/п	Раздел/тема	Занятия лекционного типа, (акад.часов)	Занятия семинарского типа, (акад. часов)		Самостоя-тельная работа, (акад. часов)	Формируемые компетенции
			Семинары и/или практические занятия	Лабораторные работы		
1	Раздел 1. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ					ОПК–8
1.1	Основы алгоритмизации	2			4	
1.2	Языки программирования	2			4	
1.3	Язык C# и технология .Net Framework	2		6	10	
1.4	Базовые типы данных и операторы	4		6	10	
1.5	Алгоритмы разветвленной структуры	4		10	10	
1.6	Алгоритмы циклической структуры	6		12	10	
1.7	Массивы	4		10	10	
1.8	Статические методы	4		10	20	
1.9	Основы объектно-ориентированного программирования	4		10	20	
1.10	Обработка текста	4		8	10	
	Итого в семестр:	**36**		**72**	**108**	
2	Раздел 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ					ОПК–8
2.1	Перегрузка операторов	4		8	12	
2.2	Свойства и индексаторы	4		8	14	
2.3	Наследование	6		10	14	
2.4	Абстрактные классы и интерфейсы	6		8	14	
2.5	Перечисления и структуры	2		8	12	
2.6	Коллекции	2		10	14	
2.7	Обработка исключений	6		10	14	
2.8	Делегаты и события	6		10	14	
	Итого в семестр:	**36**		**72**	**108**	
3	Раздел 3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ					ОПК–8
3.1	Приложения с графическим интерфейсом	6		16	72	
3.2	Дополнительные возможности языка	6		14	12	
3.3	Указатели	8		14	8	
3.4	Многопоточное программирование	8		14	8	
3.5	Обобщенные типы	8		14	8	
	Итого в семестр:	**36**		**72**	**108**	
	Всего:	**108**		**216**	**324**	

б) заочная форма

№ п/п	Раздел/тема	Занятия лекционного типа, (акад.часов)	Занятия семинарского типа, (акад. часов)		Самостоя-тельная работа, (акад. часов)	Формируемые компетенции
			Семинары и/или практические занятия	Лабораторные работы		
1	Раздел 1. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ					ОПК–8
1.1	Основы алгоритмизации	2			34	
	Итого в семестр:	**2**			**34**	
1.2	Языки программирования	0,5			10	
1.3	Язык C# и технология .Net Framework	0,5		2	10	
1.4	Базовые типы данных и операторы	0,5		2	20	
1.5	Алгоритмы разветвленной структуры	0,5		2	20	
1.6	Алгоритмы циклической структуры	1		2	30	
1.7	Массивы	1		2	20	
1.8	Статические методы	1		2	30	
1.9	Основы объектно-ориентированного программирования	2		4	30	
1.10	Обработка текста	1		2	20	
	Итого в семестр:	**8**		**18**	**190**	
2	Раздел 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ					ОПК–8
2.1	Перегрузка операторов	1		2	24	
2.2	Свойства и индексаторы	1		2	24	
2.3	Наследование	2		2	24	
2.4	Абстрактные классы и интерфейсы	2		2	24	
2.5	Перечисления и структуры	0,5		2	20	
2.6	Коллекции	0,5		2	24	
2.7	Обработка исключений	1		2	24	
2.8	Делегаты и события	2		4	24	
	Итого в семестр:	**10**		**18**	**188**	
3	Раздел 3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ					ОПК–8
3.1	Приложения с графическим интерфейсом	2		4	108	
3.2	Дополнительные возможности языка	2		2	11	
3.3	Указатели	2		4	11	
3.4	Многопоточное программирование	2		4	11	
3.5	Обобщенные типы	2		4	11	
	Итого в семестр:	**10**		**18**	**152**	
	Всего:	**30**		**54**	**564**	

Программой дисциплины «Программирование» предусмотрены занятия лекционного типа, занятия семинарского типа и самостоятельная работа обучающихся.

На занятиях семинарского типа выполняются лабораторные работы.

Самостоятельная работа предполагает изучение обучающимися теоретического курса, курсовое проектирование.

Примерный перечень тем курсовой работы приводится в Фонде оценочных средств для проведения текущего контроля и промежуточной аттестации по дисциплине (ФОС), представленном в приложении к рабочей программе.

Для запланированных видов занятий разработаны учебно-методические материалы, которые включены в состав электронного учебно-методического комплекса дисциплины (ЭУМКД) по дисциплине «Программирование» [8].

Практическая подготовка при реализации дисциплины «Программирование» организуется путем проведения лабораторных работ, предусматривающих участие

обучающихся в выполнении отдельных элементов работ, связанных с будущей профессиональной деятельностью.

5.2. Занятия лекционного типа

№ темы	Раздел/тема дисциплины	Краткое содержание лекционного занятия
1	Раздел 1. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ	
1.1	Основы алгоритмизации	Этапы решения задачи на ЭВМ. Алгоритмы. Способы описания алгоритмов
1.2	Языки программирования	Основные элементы языка программирования. Ассемблеры. Языки высокого уровня. Компиляторы и интерпретаторы. Классификация языков программирования
1.3	Язык C# и технология .Net Framework	Язык программирования C#. Программное обеспечение. Структура программы. Простейшая консольная программа
1.4	Базовые типы данных и операторы	Понятие переменной. Базовые типы данных. Инициализация переменных. Блочное объявление переменных. Константы. Литералы. Преобразование типов. Арифметические операторы. Математические методы. Форматированный вывод. Управляющие символы. Простейшая целочисленная арифметика
1.5	Алгоритмы разветвленной структуры	Логические выражения. Приоритет операций. Тернарный оператор. Оператор ветвления IF. Вложенные операторы IF. Оператор Switch
1.6	Алгоритмы циклической структуры	Оператор цикла for. Управление циклическими операторами. Примеры задач. Вложенные циклы for. Оператор цикла While. Оператор цикла do-while. Перехват исключений
1.7	Массивы	Основные понятия. Объявление одномерного массива. Инициализация одномерного массива. Операции с массивами. Сортировка элементов массива. Цикл по массиву. Объявление двумерного массива. Инициализация двумерного массива. Обработка элементов массива
1.8	Статические методы	Общие сведения о методах. Описание метода. Вызов метода. Перегрузка статических методов. Массив как аргумент метода. Массив как результат метода. Механизмы передачи аргументов методу. Рекурсия
1.9	Основы объектно-ориентированного программирования	Базовые принципы ООП. Понятие класса и объекта. Описание класса. Создание объекта. Конструктор. Деструктор. Статические члены класса. Ключевое слово this
1.10	Обработка текста	Класс String. Класс String. Операции с текстовыми объектами. Методы для работы с текстом. Метод ToString()
2	Раздел 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	
2.1	Перегрузка операторов	Операторные методы. Перегрузка операторов. Перегрузка операторов приведения типов
2.2	Свойства и индексаторы	Понятие свойства. Использование свойств. Понятие индексатора. Использование индексатора. Индексаторы без базового массива. Двумерные индексаторы
2.3	Наследование	Понятие наследования. Доступ к членам класса и наследование. Наследование и конструкторы. Наследование и сокрытие имен. Создание многоуровневой иерархии. Ссылки на базовый класс и объекты производных классов. Ссылки на базовый класс и объекты производных классов. Виртуальные методы и их переопределение. Переопределение и замещение методов. Перегрузка методов при наследовании. Наследование свойств и индексаторов
2.4	Абстрактные классы и интерфейсы	Понятие абстрактного класса. Абстрактные свойства и индексаторы. Понятие интерфейса. Наследование интерфейсов. Интерфейсные переменные. Явная реализация членов интерфейса
2.5	Перечисления и структуры	Понятие перечисления. Инициализация перечисления. Понятие структуры
2.6	Коллекции	Понятие коллекции. Использование класса ArrayList. Использование класса List<T>
2.7	Обработка исключений	Принципы обработки исключений. Использование конструкции try-catch. Основные классы исключений. Обработка нескольких catch-блоков. Вложение блоков try. Использование блока finally. Генерация

		исключений. Пользовательские классы исключений. Инструкции checked и unchecked
2.8	Делегаты и события	Понятие делегата. Многократная адресация. Анонимные методы. Лямбда-выражения. События
3	Раздел 3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ	
3.1	Приложения с графическим интерфейсом	Основные понятия. Создание приложений с графическим интерфейсом. Модель и представление
3.2	Дополнительные возможности языка	Работа с файлами. Работа с датой и временем. Использование пространства имен
3.3	Указатели	Знакомство с указателями. Адресная арифметика. Указатели на экземпляр структуры. Инструкция fixed. Указатели и массивы. Указатели и текст. Массив указателей. Примеры
3.4	Многопоточное программирование	Класс Thread и создание потоков. Фоновые потоки. Операции с потоками. Заполнение массива потоками. Массив объектных переменных для потоков. Создание потока в потоке. Вычисление суммы
3.5	Обобщенные типы	Обобщенные методы. Перегрузка обобщенных методов. Нестатические обобщенные методы. Обобщенные классы. Обобщенные структуры. Обобщенные интерфейсы. Обобщенные классы и наследование. Обобщенные делегаты. Ограничения на параметры типа

5.3. Занятия семинарского типа

5.3.1. Лабораторные работы

№ темы	Раздел/тема дисциплины	Наименование и объем лабораторной работы (очная/заочная формы)	Краткое содержание лабораторной работы
1	Раздел 1. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ		
1.3	Язык C# и технология .Net Framework	Создание простейших консольных и диалоговых программ в C# (6/2 часа)	Знакомство с алгоритмом создания проекта в C#, структурой программы, понятием метода, понятием переменной. Создание простейшей консольной программы. Создание простейшей программы с диалоговым окном. Настройка вида диалогового окна. Создание окна с полем ввода. Консольный ввод. Форматированный вывод
1.4	Базовые типы данных и операторы	Элементы языка. Базовые типы данных и операторы (6/2 часа)	Знакомство с объявлением и инициализацией переменных в C#, оператором присваивания, вычислением арифметических выражений, использованием операторов / и %
1.5	Программирование алгоритмов циклической структуры	Алгоритмы разветвленной структуры (10/2 часа)	Изучение типа данных bool и особенностей его использования. Изучение операций сравнения и логических операций, умение составлять с их помощью сложные логические выражения. Использование тернарного оператора. Изучение оператора If и оператора Switch
1.6	Алгоритмы циклической структуры	Оператор цикла for (6/1 часа)	Изучение базовой алгоритмической конструкции «цикл». Изучение оператора цикла for, вложенных циклов, цикла foreach
		Операторы цикла while, do while (6/1 часа)	Изучение базовой алгоритмической конструкции «цикл». Изучение оператора цикла с предусловием (while), цикла с постусловием (do while)
1.7	Массивы	Одномерные массивы (4/1 часа)	Овладение практическими навыками работы с одномерными массивами, особенностями их ввода и вывода, приобретение дальнейших навыков по организации программ циклической структуры с использованием приемов программирования

		Двумерные массивы (6/1 часа)	Овладение практическими навыками работы с двумерными массивами, особенностями их ввода и вывода, приобретение дальнейших навыков по организации программ циклической структуры с использованием приемов программирования
1.8	Статические методы	Статические методы (8/2,5 часов)	Изучение правил описания и использования статических методов на языке C#, изучение вариантов передачи аргументов, особенностей использования локальных и глобальных переменных программы
		Рекурсия (2/0,5 часа)	Изучение правил описания и использования рекурсии на языке C#
1.9	Основы объектно-ориентированного программирования	Использование объектов (10/4 часа)	Знакомство с алгоритмом создания и использования объектов в C#. Описание простейшего класса и создание объектов. Присваивание объектов. Закрытые члены класса и перегрузка методов. Использование конструктора. Использование деструктора. Статические поля и методы. Использование ключевого слова this
1.10	Обработка текста	Обработка текста (8/2 часа)	Знакомство с алгоритмами создания и использования текстовых объектов в C#. Сравнение строк. Поиск по тексту. Вставка и замещение текста
2	Раздел 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ		
2.1	Перегрузка операторов	Перегрузка операторов (8/2 часа)	Изучение вопросов, связанных с перегрузкой операторов в C#. Перегрузка арифметических операторов. Перегрузка операторов сравнения. Перегрузка логических операторов. Перегрузка операторов приведения типов
2.2	Свойства и индексаторы	Свойства и индексаторы (8/2 часа)	Изучение вопросов, связанных с использованием свойств и индексаторов в C#
2.3	Наследование	Наследование (10/2 часа)	Изучение вопросов, связанных с использованием наследования – фундаментального механизма объектно-ориентированного программирования
2.4	Абстрактные классы и интерфейсы	Абстрактные классы и интерфейсы (8/2 часа)	Изучение вопросов, связанных с созданием и использованием абстрактных классов и интерфейсов
2.5	Перечисления и структуры	Перечисления и структуры (8/2 часа)	Изучение вопросов, связанных с созданием и использованием перечислений и структур
2.6	Коллекции	Коллекции (10/2 часа)	Создание и использование коллекций. Использование класса ArrayList. Использование класса List<T>
2.7	Обработка исключений	Обработка исключений (10/2 часа)	Обработка исключительный ситуаций. Использование конструкции try-catch. Основные классы исключений. Обработка нескольких catch-блоков. Вложение блоков try. Использование блока finally. Генерация исключений. Пользовательские классы исключений. Инструкции checked и unchecked
2.8	Делегаты и события	Делегаты и события (10/4 часа)	Создание и использование делегатов. Многократная адресация. Анонимные методы. Лямбда-выражения. Использование событий
3	Раздел 3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ		
3.1	Приложения с графическим интерфейсом	Разработка приложений с графическим интерфейсом (16/4 часа)	Создание пустого окна. Окно и обработка событий. Использование компонентов. Создание простейшей программы для просмотра рисунков. Ввод и вывод данных. Создание новых компонентов. Модель и представление.
3.2	Дополнительные возможности языка	Изучение дополнительных возможностей языка (14/2 часа)	Работа с файлами. Работа с датой и временем. Использование пространства имен. Упражнения и деловые игры, направленные на формирование команды. Создание командных проектов, решение задач в команде
3.3	Указатели	Указатели и массивы (14/4 часа)	Создание указателей. Индексирование указателей. Операции с указателями. Создание указателей на экземпляр структуры. Использование инструкции fixed.

			Указатели и массивы. Указатели и текст. Многоуровневая адресация
3.4	Многопоточное программирование	Использование потоков (14/4 часа)	Создание потоков. Создание дочерних потоков. Координация завершения потоков. Поток на основе экземпляра делегата. Фоновый поток. Операции с потоками. Синхронизация потоков. Заполнение массива потоками. Массив объектных переменных для потоков. Создание потока в потоке
3.5	Обобщенные типы	Обобщенные типы (14/4 часа)	Создание обобщенных статических методов. Создание обобщенных классов со свойствами обобщенного типа. Создание обобщенных классов с индексаторами обобщенного типа. Создание цепочки объектов на основе обобщенного класса
	Всего:	**216/54**	

5.4. Занятия в форме практической подготовки

№ темы	Раздел/тема дисциплины	Тип занятия	Наименование и объем занятия, часа(ов) очная/заочная
	Раздел 3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ		
3.5	Приложения с графическим интерфейсом	Лабораторная работа	Разработка приложений с графическим интерфейсом (16/4)
3.6	Дополнительные возможности	Лабораторная работа	Изучение дополнительных возможностей языка (14/2)
	Всего:		**30/6**

6. Оценочные материалы для проведения текущего контроля и промежуточной аттестации обучающихся по дисциплине

Оценочные материалы для текущего контроля успеваемости и промежуточной аттестации по итогам освоения дисциплины «Программирование» сформированы в виде фонда оценочных средств (ФОС) и представлены в приложении к рабочей программе.

7. Учебно-методическое обеспечение дисциплины

7.1. Рекомендуемая литература

№ п/п	Наименование	Электронный адрес	Кол-во экз.
	7.1.1. Основная литература		
1	Казанский, А. А. Программирование на Visual C# : учебное пособие для вузов / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023.	https://urait.ru/bcode/512404	
2	Вирт, Н. Алгоритмы и структуры данных. Новая версия для Оберона + CD [Текст] / Н. Вирт. – 2-е изд., испр. - М.: ДМК Пресс, 2012. – 272 с.		5
3	Гагарина, Л. Г. Алгоритмы и структуры данных [Текст]: учеб. пособие / Л. Г. Гагарина, В. Д. Колдаев. - М.: Финансы и статистика, 2009. - 303 с.		20
4	Кудрина, Е. В. Основы алгоритмизации и программирования на языке C# : учебное пособие для вузов / Е. В. Кудрина, М. В. Огнева. — Москва : Издательство Юрайт, 2023. — 322 с.	https://urait.ru/bcode/517285	
5	Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2022. — 206 с.	https://urait.ru/bcode/490369	
	7.1.2. Дополнительная литература		

6	Вычислительная техника и программирование. Практикум по программированию [Текст] : практ. пособие / В. Е. Алексеев, А. С. Ваулин, Г. Б. Петрова ; ред. А. В. Петров. - М. : Высш.шк., 1991. - 400 с.		46
7	Зыков, С. В. Программирование. Объектно-ориентированный подход : учебник и практикум для вузов / С. В. Зыков. — Москва : Издательство Юрайт, 2023. — 155 с.	https://urait.ru/bcode/512425	
8	Программирование [Электронный ресурс]: электронный учеб.-метод. комплекс / сост. П.А. Егармин. – Лесосибирск, 2017	http://www.lfsibgu.ru/elektronnyj-katalog	

7.2. Перечень современных профессиональных баз данных и информационных справочных систем, необходимых для освоения дисциплины

№ п/п	Наименование
1.	Научно-техническая библиотека филиала СибГУ в г. Лесосибирске : [сайт]. – Лесосибирск, 2004 – . – http://lfsibgu.ru/elektronnyj-katalog. – Текст : электронный.
2.	Лань : электронно-библиотечная система издательства : [сайт]. – Москва, 2010 – . – URL: http://e.lanbook.com – Режим доступа: по подписке. – Текст : электронный.
3.	ЮРАЙТ : образовательная платформа : [сайт]. – Москва, 2013 – URL: https://urait.ru/ – Режим доступа: по подписке. – Текст : электронный.
4.	IPR SMART : цифровой образовательный ресурс: [сайт] . – Москва, 2021 – . – URL: https://www.iprbookshop.ru/ – Режим доступа: по подписке. – Текст : электронный.
5.	Сервер электронно-дистанционного обучения СибГУ им. М. Ф. Решетнева : [электрон. образоват. ресурс для студентов всех форм обучения] : [сайт]. – URL: https://dl.sibsau.ru – Режим доступа: для авториз. пользователей. – Текст : электронный.

7.3. Методические указания для обучающихся по освоению дисциплины

Программой дисциплины «Программирование» предусмотрены занятия лекционного типа, занятия семинарского типа (лабораторные работы) и самостоятельная работа обучающихся.

Самостоятельная работа предполагает изучение теоретического курса и курсовое проектирование. В период освоения дисциплины для обучающихся организуются индивидуальные и групповые консультации.

При изучении дисциплины обязательным является выполнение следующих организационных требований:

– обязательное посещение всех видов аудиторных занятий;
– ведение конспекта лекций, практических занятий;
– активная работа во время занятий;
– регулярная самостоятельная работа обучающегося в соответствии с рабочей программой дисциплины и рейтинг планом;
– своевременная сдача отчетных документов;
– получение дополнительных консультаций по подготовке, оформлению и сдаче отдельных видов заданий, в случае пропусков занятий.

Самостоятельная работа обучающегося направлена на:

– стимулирование познавательного интереса;
– систематизацию и закрепление полученных теоретических знаний;
– развитие познавательных способностей, активности, самостоятельности, ответственности и организованности обучающихся;
– формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации.

Чтобы выполнить весь объем самостоятельной работы по всем осваиваемым дисциплинам, обучающемуся необходимо заниматься по 3-5 часов ежедневно. Начинать самостоятельные внеаудиторные занятия следует с первых же дней семестра, поскольку компенсировать пропущенный материал позднее без снижения качества работы и ее производительности практически невозможно.

Вид учебных занятий	Организация деятельности обучающегося
Лекция	Лекции имеют целью дать систематизированные знания об изучаемой предметной области. В ходе лекций преподаватель излагает и разъясняет основные, наиболее сложные понятия темы, а также связанные с ней теоретические и практические проблемы, дает рекомендации на лабораторные работы и указания на самостоятельную работу. В ходе лекций обучающимся рекомендуется: – вести конспектирование учебного материала; – обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации по их применению; – задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций. Желательно оставить в рабочих конспектах поля, на которых во внеаудиторное время можно сделать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. Для успешного овладения курсом необходимо посещать все лекции, так как тематический материал взаимосвязан между собой.
Лабораторная работа	При подготовке к лабораторным работам обучающемуся необходимо изучить методические указания по выполнению лабораторной работы, изучить основные теоретические положения по теме работы, выполнить экспериментальную часть, произвести необходимые расчеты, оценить правильность полученных результатов. Лабораторные работы выполняются подгруппами обучающихся в специализированных лабораториях. Каждую лабораторную работу обучающийся должен оформить в виде отчета, который представляется на рассмотрение преподавателя, защитить отчет, предоставив выполненные задания и ответив на контрольные вопросы.
Самостоятельная работа (изучение теоретической части курса)	При изучении дисциплины не все вопросы рассматриваются на лекциях и практических занятиях, часть вопросов рекомендуется преподавателем для самостоятельного изучения. При самостоятельном изучении и проработке теоретического курса необходимо повторить законспектированный во время лекции материал и дополнить его с учетом рекомендованной литературы. Важной частью самостоятельной работы является чтение учебной и научной литературы. Основная функция учебников - ориентировать обучающихся в системе знаний, умений и навыков, которые должны быть усвоены по данной дисциплине будущими специалистами. Поиск ответов на вопросы и выполнение заданий для самостоятельной работы позволяет расширить и углубить знания по курсу, применить теоретические знания в решении задач практического содержания, закрепить изученное ранее. Уровень усвоения материала может быть оценен при ответах на контрольные вопросы для самопроверки по соответствующим темам и разделам.
Самостоятельная работа (курсовая работа)	Выполнение курсовой работы является обязательным условием для допуска обучающегося к экзамену. Курсовая работа представляет собой изложение в письменном и графическом виде результатов теоретического анализа и практической работы обучающегося по определенной теме. Содержание курсовой работы зависит от выбранного варианта. Работа представляется преподавателю на проверку не позднее, чем за 7 дней до планируемой защиты. Защита работы проходит в форме собеседования во время консультаций (до начала экзамена), или в сроки, установленные графиком экзаменационной сессии.
Подготовка к экзамену	Подготовка к экзамену предполагает изучение рекомендуемой литературы и других источников, конспектов лекций, повторение материалов лабораторных работ.

8. Материально-техническое обеспечение дисциплины

Наименование аудитории	Назначение аудитории	Оборудование
Учебная аудитория	для проведения занятий лекционного типа, групповых и	Учебная мебель для обучающихся, доска, рабочее место преподавателя. Комплект мультимедийного оборудования.

	индивидуальных консультаций, текущего контроля и промежуточной аттестации, курсового проектирования	
		Перечень лицензионного и свободно распространяемого программного обеспечения, необходимого для освоения дисциплины: 1. Операционная система Microsoft Windows. 2. Офисный пакет Microsoft Office. 3. Браузер Google Chrome. 4. Антивирус Dr. Web. 5. Система программирования Microsoft Visual Studio.
Учебная аудитория	для проведения занятий семинарского типа (лабораторных), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, курсового проектирования	Учебная мебель для обучающихся, доска, рабочее место преподавателя.
		Компьютеры с подключением к информационно-телекоммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду СибГУ им. М.Ф. Решетнева. Перечень лицензионного и свободно распространяемого программного обеспечения, необходимого для освоения дисциплины: 1. Операционная система Microsoft Windows. 2. Офисный пакет Microsoft Office. 3. Браузер Google Chrome. 4. Антивирус Dr. Web. 5. Система программирования Microsoft Visual Studio.
Помещение для самостоятельной работы	для самостоятельной работы обучающихся	Компьютеры с подключением к информационно-телекоммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду СибГУ им. М.Ф. Решетнева

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Лесосибирский филиал федерального государственного бюджетного образовательного учреждения высшего образования «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева»

ФОНД
ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации
по дисциплине
(приложение к рабочей программе дисциплины)

ПРОГРАММИРОВАНИЕ

09.03.01 Информатика и вычислительная техника
Код Наименование

Направленность (профиль) образовательной программы
Программное обеспечение средств вычислительной техники и автоматизированных систем

Уровень высшего образования
бакалавриат

Форма обучения
очная, заочная

Красноярск 2022

Фонд оценочных средств
для проведения промежуточной аттестации

по дисциплине Программирование

1. Описание назначения и состава фонда оценочных средств

Настоящий фонд оценочных средств (ФОС) входит в состав рабочей программы дисциплины Программирование

и предназначен для оценки планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенций.

ФОС включает в себя оценочные материалы для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся в форме: экзамена, курсовой работы.

В состав ФОС входят следующие оценочные средств:

– компьютерные тесты по темам дисциплины (текущий контроль);
– устный опрос (текущий контроль);
– задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ (текущий контроль);
– задания для выполнения курсовой работы (промежуточная аттестация);
– вопросы к экзамену (промежуточная аттестации).

2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенций

Код компетенции	Содержание компетенции	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенции
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения	**ОПК-8.1.** Применяет методы проектирования программного обеспечения **ОПК-8.2.** Использует современные языки программирования для разработки алгоритмов и программ **ОПК-8.3.** Владеет методами отладки и тестирования программ	**Знать:** – основы алгоритмизации и программирования; – современные среды разработки программного обеспечения; – возможности современных и перспективных средств разработки программных продуктов; – синтаксис и основные особенности одного из современных языков программирования. **Уметь:** – составлять алгоритмы; – писать и отлаживать коды на языке программирования; – тестировать работоспособность программы; – интегрировать программные модули. **Владеть:** – языком программирования; – методами отладки и тестирования работоспособности программы

2.1. Формы контроля формирования компетенций

№	Контролируемые раздел/тема дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1	Раздел 1. ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ	ОПК-8	
1.1	Основы алгоритмизации		**Текущий контроль:** устный опрос, компьютерный тест
1.2	Языки программирования		**Текущий контроль:** устный опрос, компьютерный тест
1.3	Язык C# и технология .Net Framework		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.4	Базовые типы данных и операторы		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.5	Алгоритмы разветвленной структуры		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.6	Алгоритмы циклической структуры		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.7	Массивы		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.8	Статические методы		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.9	Основы объектно-ориентированного программирования		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
1.10	Обработка текста		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2	Раздел 2. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ	ОПК-8	
2.1	Перегрузка операторов		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2.2	Свойства и индексаторы		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2.3	Наследование		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2.4	Абстрактные классы и интерфейсы		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2.5	Перечисления и структуры		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2.6	Коллекции		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
2.7	Обработка исключений		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ

№	Контролируемые раздел/тема дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
2.8	Делегаты и события		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
3	Раздел 3. РАСШИРЕННЫЕ ВОЗМОЖНОСТИ	ОПК-8	
3.1	Приложения с графическим интерфейсом		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
3.2	Дополнительные возможности		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
3.3	Указатели		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
3.4	Многопоточное программирование		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
3.5	Обобщенные типы		**Текущий контроль:** задания для выполнения лабораторных работ и вопросы для защиты лабораторных работ
	Промежуточная аттестация		**Промежуточная аттестация по дисциплине:** вопросы к экзамену, задания для выполнения курсовой работы

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков владения, соотнесенных с установленными в программе индикаторами достижения компетенций

3.1. Устный опрос (текущий контроль), формирование компетенции ОПК-8

Устный опрос проводится выборочно в начале лекции и включает перечень вопросов по пройденному материалу. Устный опрос проводится по темам, по которым нет лабораторных работ.

Перечень вопросов для устного опроса.

1. Назовите основные этапы решения задачи на ЭВМ.
2. Что такое алгоритм? Перечислите свойства алгоритма.
3. Перечислите способы задания алгоритмов. Приведите примеры.
4. Опишите словесно алгоритм нахождения наибольшего общего делителя (НОД) двух натуральных чисел.
5. Имеется два кувшина емкостью 3 и 8 литров. Составьте словесный алгоритм, с помощью которого, пользуясь только двумя этими кувшинами, можно набрать 7 литров воды.
6. Перечислите недостатки словесных алгоритмов.
7. Приведите блок-схему алгоритма решения линейного уравнения.
8. Приведите псевдокод алгоритма решения линейного уравнения.
9. Что такое компьютерная программа, язык программирования?
10. Какие основные элементы содержит язык программирования?
11. Что такое синтаксис и семантика языка программирования?
12. Приведите примеры языков программирования низкого и высокого уровня.
13. Что такое транслятор? Какие виды трансляторов вы знаете?
14. Приведите алгоритм работы компилятора, интерпретатора. В чем разница между компиляторами и интерпретаторами?
15. Приведите классификацию языков программирования.

16. В чем заключается основная идея процедурного программирования? Приведите примеры процедурных языков программирования.
17. В чем заключается концепция объектно-ориентированного программирования? Приведите примеры объектно-ориентированных языков программирования.
18. Что такое функциональное программирование, логическое программирование?
19. Какие языки программирования относятся к языкам программирования баз данных, языкам программирования для компьютерных сетей?
20. Какие элементы включает система программирования?

3.2. Компьютерные тесты (текущий контроль), формирование компетенции ОПК-8

Подробное описание тестовых заданий содержится в сборнике тестовых заданий, который включен в состав ЭУМКД [8].

Примеры тестовых заданий по дисциплине.

1. Словесный способ описания алгоритма заключается в том, что алгоритм:

- описывается в произвольном изложении на естественном языке
- описывается с помощью геометрических фигур (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций
- описывается на естественном, частично формализованном языке, позволяющее выявить основные этапы решения задачи, перед его точной записью на языке программирования
- описывается с помощью основных конструкций языка программирования

2. Алгоритм это:

- точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время
- произвольный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за конечное время
- точный набор инструкций, описывающих порядок действий исполнителя для достижения результата решения задачи за неопределенное время
- точный набор инструкций, описывающих порядок действий исполнителя для достижения или недостижения результата решения задачи за конечное время

3. Язык программирования C# создан инженерами компании:

- Adobe systems
- IBM
- Microsoft
- Apple
- Google

4. Какой метод используются для консольного ввода?

- WriteLine()
- ReadLine()
- Show()
- Main()

5. В объектно-ориентированном программировании термин "метод" означает:

- именованный блок из команд
- шаблонный код
- тело программы
- пространство имен

6. Алгоритм разветвленной структуры – это:

- алгоритм, в котором предусмотрено разветвление указанной последовательности действий на два направления в зависимости от итога проверки заданного условия
- конструкция, реализованная в виде последовательности действий, каждое из которых выполняется ровно один раз
- алгоритм, в котором идущая подряд группа действий (шагов) может выполняться несколько раз, в зависимости от входных данных или условия задачи
- условие, состоящее из нескольких простых условий (отношений), связанных с помощью логических операций

7. Описание алгоритма на алгоритмическом языке (псевдокод) заключается в том, что алгоритм:

- описывается на естественном, частично формализованном языке, позволяющее выявить основные этапы решения задачи, перед его точной записью на языке программирования
- описывается с помощью команд языка программирования и геометрических блоков
- описывается в произвольном изложении на естественном языке
- представляется в виде геометрических фигур (блоков), имеющих определенную конфигурацию в зависимости от характера выполняемых операций

8. Результативность алгоритма (как его свойство) заключается в том, что:

- любой алгоритм должен завершаться за конечное число шагов
- алгоритм должен состоять из отдельных шагов (команд)
- алгоритм должен быть применим к разным наборам исходных данных
- каждая команда алгоритма должна однозначно определять действие исполнителя

3.3. Задания для лабораторных работ на занятиях семинарского типа и вопросы для защиты лабораторных работ (текущий контроль), формирование компетенции ОПК-8

Подробное описание лабораторных работ и вопросов для защиты лабораторных работ содержатся в Практикуме по выполнению лабораторных работ, который включен в состав ЭУМКД [8].

Примерные задания для лабораторной работы по теме «Абстрактные классы и интерфейсы».

1. Напишите программу, содержащую абстрактный базовый класс с защищенным полем, являющимся ссылкой на целочисленный массив. У класса должен быть конструктор с одним аргументом (определяет размер массива и создает его), целочисленное свойство (значение – размер массива), абстрактный метод (без аргументов, не возвращает результат) и абстрактный индексатор с целочисленным индексом (доступен для чтения и записи). В производном классе описать абстрактный метод из базового класса, чтобы он отображал в консоли содержимое массива. Абстрактный индексатор определить так, чтобы с его помощью можно было прочитать значение элемента массива и присвоить значение элементу массива.

2. Напишите программу, в которой на основе двух интерфейсов создается класс. В одном интерфейсе объявлен индексатор с символьным индексом, возвращающий целочисленное значение. В другом интерфейсе объявлен индексатор с целочисленным индексом, возвращающий символьное значение.

3. Напишите программу, содержащую базовый класс с защищенным текстовым полем. У класса должен быть конструктор с текстовым аргументом, доступное для чтения абстрактное целочисленное свойство, доступный для чтения абстрактный индексатор с целочисленным индексом, не возвращающий результат абстрактный метод с текстовым аргументом, не возвращающий результат абстрактный метод без аргументов. В производном классе: свойство результатом возвращает количество

символов в текстовом поле, значением выражения с проиндексированным объектом является код символа в тексте, метод с текстовым аргументом присваивает новое значение полю, а метод без аргументов отображает значение поля в консольном окне.

4. Напишите программу, содержащую два интерфейса. В первом интерфейсе опишите метод с символьным аргументом и целочисленным результатом, а во втором интерфейсе – метод с целочисленным аргументом и символьным результатом. Опишите класс, реализующий оба интерфейса. Проверьте работу методов, вызвав их через объектную переменную и через интерфейсные переменные (там, где это возможно).
5. Напишите программу, содержащую абстрактный базовый класс с защищенным целочисленным полем. У класса должен быть конструктор с одним аргументом (определяет значение целочисленного поля), целочисленное свойство (количество знаков в значении целочисленного поля), абстрактный метод (без аргументов, не возвращает результат) и абстрактный индексатор с целочисленным индексом (доступен для чтения и записи). В производном классе описать абстрактный метод из базового класса, чтобы он отображал в консоли значение целочисленного поля. Абстрактный индексатор определить так, чтобы с его помощью можно было вывести на экран разряд числа, записанного в целочисленное поле.
6. Напишите программу, в которой будет описан интерфейс с методом без аргументов, который возвращает результатом целое число. На основе интерфейса создайте два класса. У каждого класса должно быть целочисленное поле. В первом классе метод результатом возвращает сумму четных чисел, во втором классе метод возвращает результатом сумму нечетных чисел. Количество слагаемых в сумме определяется полем объекта, из которого вызывается метод. Проверьте работу метода, получив доступ к объекту класса через объектную переменную и через интерфейсную переменную.
7. Напишите программу, содержащую базовый класс с защищенным символьным полем. У класса должен быть конструктор с символьным аргументом, доступное для чтения абстрактное целочисленное свойство, не возвращающий результат абстрактный метод с символьным аргументом, не возвращающий результат абстрактный метод без аргументов. В производном классе: свойство результатом возвращает код символа в символьном поле, метод с символьным аргументом присваивает новое значение полю, а метод без аргументов отображает значение поля в консольном окне.
8. Напишите программу, в которой класс создается на основе двух интерфейсов. В первом интерфейсе есть целочисленное свойство, доступное для чтения и записи. Во втором интерфейсе есть текстовое свойство, доступное для чтения и записи. В каждом из интерфейсов объявлены одинаковые (с одинаковыми именами) методы, без аргументов, не возвращающие результат. В классе описать соответствующий метод, который в консольном окне отображает значения свойств.

Перечень вопросов для защиты лабораторной работы по теме «Абстрактные классы и интерфейсы».

1. Что такое абстрактный метод, абстрактный класс?
2. Перечислите особенности использования абстрактных классов, методов, свойств и индексаторов.
3. В чем преимущество использования абстрактных классов и методов по сравнению с обычными классами и методами?
4. Что такое множественное наследование? Почему в C# не используется множественное наследование?

5. Как вы понимаете, что такое интерфейс? Перечислите основные особенности использования интерфейсов.
6. Как реализуется наследование (расширение) интерфейсов?
7. Что такое интерфейсная переменная? Перечислите особенности ее использования.
8. В чем заключается явная реализация членов интерфейса?

3.4. Задания для выполнения курсовой работы (промежуточная аттестация), формирование компетенции ОПК-8

Курсовая работа выполняется студентами по индивидуальным заданиям. Индивидуальные задания содержатся в учебном пособии по курсовому проектированию, которое включено в состав ЭУМКД [8].

Пример задания для выполнения курсовой работы. Деканат вуза располагает следующей информацией о результатах текущей сессии студентов очной формы обучения:

– фамилия;
– имя;
– номер курса;
– наименование группы;
– информация о сдаче зачетов (сдал/ не сдал)
– экзаменационная оценка за 1 предмет;
– экзаменационная оценка за 2 предмет;
– экзаменационная оценка за 3 предмет.

Сведения о результатах сессии представлены в ведомости, о которой известна следующая информация:

– месяц формирования ведомости;
– количество студентов.

Необходимо:

1. Построить модель предметной области.
2. Построить диаграмму классов проектирования.
3. Разработать приложение с графическим интерфейсом в соответствии с диаграммой классов проектирования, реализующее следующие задачи:
   – для каждого студента определить назначена ли ему стипендия (все зачеты сданы, нет оценок 2 и 3);
   – определить количество студентов, получающих стипендию в текущем месяце;
   – вывести на экран список отличников (все зачеты сданы, все оценки 5);
   – вывести на экран список неуспевающих студентов).
4. В программе предусмотреть:
   – организацию диалога с пользователем во время работы программы;
   – генерацию числовых данных с помощью счетчика случайных чисел;
   – сохранение данных в файл.

Примерный перечень тем для курсового проектирования по дисциплине «Программирование»

№ п/п	Наименование темы курсовой работы
1.	Разработка приложения с графическим интерфейсом «Деканат вуза»
2.	Разработка приложения с графическим интерфейсом «Библиотека»
3.	Разработка приложения с графическим интерфейсом «Автосалон»
4.	Разработка приложения с графическим интерфейсом «Приемная комиссия»
5.	Разработка приложения с графическим интерфейсом «Салон сотовой связи»
6.	Разработка приложения с графическим интерфейсом «Магазин бытовой техники»
7.	Разработка приложения с графическим интерфейсом «Бухгалтерия предприятия»

8.	Разработка приложения с графическим интерфейсом «Отдел кадров»

3.5. Вопросы экзамену (промежуточная аттестация), формирование компетенции ОПК-8

1. Этапы решения задач на ЭВМ (основные этапы, понятие модели, пример решения задачи на ЭВМ).
2. Алгоритм. Способы описания алгоритмов (понятие алгоритма, свойства алгоритмов, словесный способ описания алгоритмов, графический способ, описание алгоритма с помощью псевдокода, примеры).
3. Понятие компьютерной программы, языка программирования. Основные элементы языка программирования (символы, лексемы, выражения, операторы).
4. Языки программирования низкого и высокого уровня (компьютерная язык программирования, алфавит, синтаксис, семантика, операторы, языки высокого и низкого уровня, трансляторы, алгоритмы работы компиляторов и интерпретаторов, достоинства и недостатки компиляторов и интерпретаторов).
5. Классификация языков программирования (процесс создания программы, состав системы программирования, классификация языков программирования, примеры).
6. В чем заключаются особенности использования языка C# и платформы .NET Framework? Какие интегрированные среды для работы с языком C# Вы знаете? Охарактеризуйте каждую из них. Подробно опишите структуру программы на языке C#.
7. Как Вы понимаете, что такое метод? Какие методы Вы знаете? Приведите примеры. Как Вы понимаете, что такое пространство имен? Какие методы используются для консольного ввода и консольного вывода? Что такое форматированный вывод, приведите примеры. Какой метод используется для отображения диалогового окна, для окна с полем ввода? Опишите назначение метода Parse().
8. Что такое переменная, оператор присваивания? Каким образом объявляются и инициализируются переменные? Перечислите особенности использования оператора присваивания.
9. Перечислите базовые типы данных в C#. Каким образом в C# происходит явное и неявное преобразование типов? Приведите примеры. Что такое литералы, управляющие символы?
10. Какие арифметические операторы вы знаете? Назовите особенности их использования в C#. Перечислите основные математические методы в C#. Перечислите особенности использования операторов / и %.
11. Приведите примеры таблиц истинности для операций дизъюнкции, конъюнкции, отрицания. К какому типу данных всегда принадлежит результат проверки логического выражения? Перечислите основные операции отношения и основные логические операции в C#. Как определяется порядок выполнения операций в составе логических выражений?
12. Приведите синтаксис и блок-схему условного оператора. В каком случае условные операторы называются вложенными? Приведите синтаксис и блок-схему оператора выбора.
13. Что такое цикл? Обоснуйте необходимость его применения. Приведите синтаксис оператора цикла FOR. Подробно опишите порядок команд при выполнении оператора цикла FOR. Возможно ли использование оператора цикла FOR без блоков? В каких случаях при использовании цикла FOR можно опустить фигурные скобки {}? Что такое вложенные циклы (приведите примеры)?
14. Приведите синтаксис и блок-схему оператора цикла с предусловием и постусловием. В чем отличие операторов цикла WHILE и DO-WHILE? Когда предпочтительнее использовать цикл WHILE? Каким образом с помощью цикла DO-WHILE в программе организовать проверку правильности ввода данных? Какие действия реализует инструкция BREAK? Какие действия реализует инструкция CONTINUE? Каким образом в C# реализован перехват исключений?

15. Что такое массив? В каких случаях его применение оправдано? Что такое индекс? Каким требованиям он должен удовлетворять? В чем разница между одномерным и двумерным массивом? Каким образом осуществляется реализация одномерного массива? Приведите примеры. В каком случае массивы могут участвовать в операторе присваивания без указания индекса (A=B)?

16. Перечислите способы задания значений элементов одномерного массива. Приведите примеры. Как поменять местами два элемента в одномерном массиве массиве? Как осуществить стандартные алгоритмы обработки значений элементов в одномерном массиве (достаточно привести фрагмент обработки элементов массива): нахождение среднего арифметического элементов массива; поиск минимального элемента в массиве.

17. В чем заключается принцип сортировки одномерного массива методом «пузырька»? Каким образом осуществить случайную генерацию символов в символьном одномерном массиве?

18. Что такое двумерный массив? Каким образом осуществляется реализация двумерного массива? Приведите примеры. Размерность массива определяется количеством индексов, которые нужно указать для идентификации элемента массива. С помощью какого свойства можно определить размерность массива? Приведите пример инициализации двумерного массива.

19. Каким образом описать статический метод в программе? Для чего в теле метода используется инструкция return? Какая переменная называется локальной? Когда выделяется место в памяти для локальной переменной? Что такое перегрузка метода? В чем заключаются особенности передачи методу массива в качестве аргумента? Что такое система сборки мусора? Какие существуют механизмы передачи аргументов методу?

20. Каким образом описать статический метод в программе? Для чего в теле метода используется инструкция return? Какая переменная называется локальной? Когда выделяется место в памяти для локальной переменной? Что такое перегрузка метода? В чем заключаются особенности передачи методу массива в качестве аргумента? Что такое система сборки мусора? Какие существуют механизмы передачи аргументов методу?

21. Что такое объектно-ориентированное программирование? Перечислите основные принципы объектно-ориентированного программирования. Что такое класс? Что такое объект? В чем принципиальное различие между ними? Как объявляется класс объектов в программе, поля и методы класса?

22. Как объявляется переменная для работы с объектом некоторого класса? Что в ней хранится? Что такое конструктор? Что такое деструктор? Что такое точечная нотация? Как она используется при работе с объектами? Как можно задать начальные значения для полей объекта? Приведите примеры использования ключевого слова this.

23. Каким образом реализуются текстовые значения и текстовые литералы в C#? Перечислите способы создания текстовых объектов. С помощью какого свойства можно узнать размер текста. Можно ли в программе обратиться к отдельному элементу текста? Каким образом происходит сравнение текстовых строк в C#?

24. Перечислите методы для работы с текстовыми значениями. Назначение метода ToString(). Что такое переопределение метода? Приведите пример переопределения метода ToString().

25. Что такое перегрузка операторов, операторный метод? Перечислите правила описания операторного метода. Какие операторы можно перегружать? Какие операторы перегружаются в паре? Что такое приведение (преобразование) типов?

26. Приведите примеры явного и неяного преобразования типов. Перечислите правила перегрузки операторов приведения типов.

27. Что такое свойство? Каким образом выполняется обращение к свойству? Приведите общую форму описания свойства. Что такое аксессоры и для чего они используются (подробно опишите назначение get-аксессора и set-аксессора)? Что такое индексатор? Для чего он используется?

28. Приведите общую форму описания индексатора. Подробно опишите назначение get-аксессора и set-аксессора при работе с индексаторами. Что такое перегрузка индексатора?

29. Что такое наследование? Какие приемущества дает наследование? Перечислите основные особенности наследования. Какие классы при наследовании называются базовыми, а какие производными? Приведите синтаксис производного класса при наследовании. В чем заключается влияние уровня доступа члена класса на его наследование?

30. Опишите особенности использования конструкторов в производных и базовых классах при наследовании. Опишите механизм замещения (сокрытия имен) полей и методов при наследовании. Что такое многоуровневая иерархия?

31. Возможно ли использовать объектные переменные базового класса при работе с объектами производного класса? Что такое виртуальный метод? Перечислите особенности использования виртуальных методов. В чем отличие между переопределением виртуальных методов и замещением методов? Каким образом осуществляется перегрузка методов при наследовании? Каким образом происходит наследование свойств и индексаторов?

32. В чем способ выполнения консольных программ отличается от способа выполнения программ с графическим интерфейсом? Что такое компоненты? Зачем они нужны? Как можно создать новый компонент? Для чего необходимо разделять программу на модель и интерфейс? Что обычно относят к модели, а что к представлению?

33. Что такое RAD-программирование? Перечислите этапы разработки программы в RAD-среде. На примере диалогового окна MessageBox поясните назначение аргументов метода, используемого для обработки событий графических компонентов.

34. Что такое абстрактный метод, абстрактный класс? Перечислите особенности использования абстрактных классов, методов, свойств и индексаторов. В чем преимущество использования абстрактных классов и методов по сравнению с обычными классами и методами?

35. Что такое множественное наследование? Почему в C# не используется множественное наследование? Как вы понимаете, что такое интерфейс? Перечислите основные особенности использования интерфейсов. Как реализуется наследование (расширение) интерфейсов? Что такое интерфейсная переменная? Перечислите особенности ее использования. В чем заключается явная реализация членов интерфейса?

36. Что такое делегат, экземпляр делегата? Каким образом происходит объявление делегата, инициализация делегата и вызов метода с помощью делегата? Что такое многократная адресация? Каким образом добавляются и удаляются ссылки на метод экземпляру делегата?

37. Что такое анонимный метод? Приведите синтаксис анонимного метода. Каким образом осуществляется передача аргументов анонимному методу?

38. Что такое лямбда-выражения? Приведите синтексис лямбда-выражения. В чем заключаются особенности использования лямбда-выражений?

39. Что такое событие, обработчик события? Для чего необходимы события? Опишите принцип действия событий, порядок обработки события. В чем заключается особенность использования обычных методов в качестве обработчиков события?

40. Что такое аксессоры событий? Приведите расширенный синтаксис объявления события. В чем заключается особенность применения анонимных методов и лямбда-выражений вместе с событиями?

41. Что такое перечисление? Каким образом объявляется перечисление? Как получить доступ к элементу перечисления? Как присвоить новое значение целочисленной константе в перечислении? Какие типы данных могут использоваться для целочисленных констант в перечислении? Поясните, в чем заключается назначение перечисления?

42. В чем разница между переменными ссылочного типа и типа значения? Приведите примеры. Что такое структура? Как объявляется структура? Какими способами можно создать экземпляр структуры? Перечислите особенности использования структур. В чем отличие структур от классов? Поясните, в чем заключается назначение структуры?

43. Что такое коллекция? Перечислите особенности использования коллекций и укажите их назначение. Что такое динамический массив? В чем преимущества динамических массивов перед статическими?

44. Каким образом реализуется одномерный массив через класс ArrayList, через класс List<T>? Какие методы и свойства классов ArrayList и List<T> вы знаете? В чем разница между коллекциями, созданными на базе классов ArrayList и List<T>? Каким образом реализуется двумерный массив через класс List<T>?

45. Каким образом в C# происходит обработка исключений? В чем заключается назначение конструкции try-catch? Перечислите варианты использования конструкции try-catch.Каким образом получить информацию (тип, описание, источник, метод, место) о возникшей ошибке?

46. Какие основные классы исключений вы знаете? Каким образом осуществляется обработка нескольких catch-блоков? Для чего в программе при обработке исключений используются вложенные блоки try? Для чего при обработке исключений используется блок finally?

47. Каким образом можно сгенерировать исключение вручную, сгенерировать повторное исключение? Как в C# создать пользовательский класс исключения? Для чего предназначены инструкции checked и unchecked?

4. Описание показателей, критериев, шкал оценивания планируемых результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенций

4.1. Показатели и критерии оценивания устного ответа

Оценка	Показатели оценивания	Критерии оценивания
«5» (отлично)	Полнота и правильность ответов на вопросы	Содержание ответа соответствует теме вопроса. Продемонстрировано уверенное владение понятийно-терминологическим аппаратом дисциплины, отсутствуют ошибки в употреблении терминов. Показано умелое использование категорий и терминов дисциплины в их ассоциативной взаимосвязи. Обучающимся продемонстрировано умение аргументировано излагать собственную точку зрения. Видно уверенное владение освоенным материалом, изложение сопровождено адекватными примерами из практики. Ответ четко структурирован и выстроен в заданной логике. Части ответа логически взаимосвязаны
«4» (хорошо)		Содержание ответа в целом соответствует теме вопроса. Продемонстрировано владение понятийно-терминологическим аппаратом дисциплины, присутствуют незначительные ошибки в употреблении терминов, не искажающие смысла. Показано умелое использование категорий и терминов дисциплины в их ассоциативной взаимосвязи. Продемонстрировано умение аргументировано излагать собственную точку зрения. Изложение отчасти сопровождено адекватными примерами из практики. Ответ в достаточной степени структурирован и выстроен в заданной логике без нарушений общего смысла. Части ответа логически взаимосвязаны
«3» (удовлетворительно)		Содержание ответа в целом соответствует теме вопроса. Продемонстрировано удовлетворительное знание материала, есть ошибки (25–30%). Продемонстрировано достаточное владение понятийно-терминологическим аппаратом дисциплины, есть ошибки в употреблении и трактовке терминов. Ошибки в использовании категорий и терминов

		дисциплины в их ассоциативной взаимосвязи. Нет собственной точки зрения либо она слабо аргументирована. Примеры, приведенные в ответе в качестве практических иллюстраций, в малой степени соответствуют изложенным теоретическим аспектам. Ответ плохо структурирован, нарушена заданная логика. Части ответа разорваны логически, нет связок между ними
«2» (неудовлетворительно)		Содержание ответа не соответствует теме вопроса или соответствует ему в очень малой степени. Продемонстрировано крайне низкое (отрывочное) знание материала, много ошибок – практически все данные либо искажены, либо неверны. Продемонстрировано крайне слабое владение понятийно-терминологическим аппаратом дисциплины, присутствуют многочисленные ошибки в употреблении терминов. Показаны неверные ассоциативные взаимосвязи категорий и терминов дисциплины. Отсутствует аргументация изложенной точки зрения, нет собственной позиции. Отсутствуют примеры из практики либо они неадекватны

4.2. Показатели и критерии оценивания задания в тестовой форме

Код	Вид оценочного средства	Критерии	Балл	Максимальный балл 5 – минимальный балл 1
Т.1	Тестовое задание №	выставляется студенту если 86-100% тестовых вопросов/заданий выполнено правильно	5	**5 – 3**
		выставляется студенту если 70-85% тестовых задач/заданий выполнено правильно	4	
		выставляется студенту если 50-69% тестовых задач/заданий выполнено правильно	3	
		при ответе студента менее чем на 60% вопросов, тестовое задание не зачитывается и у студента образуется долг, который должен быть закрыт в течении семестра или на зачетной неделе	н/з	

Минимальный балл, который необходимо набрать для зачета, равен 3.

4.3. Показатели и критерии оценивания заданий для лабораторных работ и вопросов для защиты лабораторных работ

Оценка	Показатели оценивания	Критерии оценивания
«5» (отлично)	Качество выполнения всех заданий лабораторных работ; полнота и правильность ответов на контрольные вопросы; оформление в соответствии с требованиями, самостоятельность выполнения, сдача лабораторных работ в установленные сроки.	Выполнены без замечаний все задания лабораторных работ; даны полные правильные ответы на контрольные вопросы; лабораторные работы выполнены самостоятельно, сданы в срок, оформлены в соответствии с требованиями
«4» (хорошо)		Задания лабораторных работ выполнены с несущественными замечаниями; недостаточно полные ответы на контрольные вопросы; лабораторные работы выполнены самостоятельно, сданы в срок, оформлены в соответствии с требованиями
«3» (удовлетворительно)		Задания лабораторных работ выполнены с существенными замечаниями, устраненными во время контактной работы с преподавателем; ошибки в ответах на контрольные вопросы; лабораторные работы выполнены с нарушениями графика, в оформлении работ есть недостатки; работы выполнены самостоятельно
«2» (неудовлетворительно)		Часть лабораторных работ или все работы выполнены из фрагментов работ других авторов и носят несамостоятельный характер; задания выполнены не

		полностью или неправильно; оформление работ не соответствует требованиям

4.4. Показатели и критерии оценивания устного (письменного) ответа на экзамене

Оценка	Показатели оценивания	Критерии оценивания
«5» (отлично, зачтено)	Знание программного материала, владение понятийным аппаратом, последовательность, логичность и стиль изложения, адекватность иллюстраций, умение анализировать классифицировать, обобщать, конкретизировать и систематизировать изученный материал.	Содержание ответа соответствует заданному вопросу. В ответе отражены все дидактические единицы, предусмотренные заданием. Продемонстрировано знание фактического материала, отсутствуют фактические ошибки. Продемонстрировано уверенное владение понятийно-терминологическим аппаратом дисциплины (уместность употребления, аббревиатуры, толкование и т.д.), отсутствуют ошибки в употреблении терминов. Обучающийся самостоятельно демонстрирует уверенное владение освоенным материалом, изложение сопровождает адекватными иллюстрациями (примерами). Ответ четко структурирован, части ответа логически взаимосвязаны. Обучающийся умеет анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал.
«4» (хорошо, зачтено):		Содержание ответа в целом соответствует заданному вопросу. Продемонстрировано знание фактического материала, встречаются несущественные фактические ошибки. Продемонстрировано владение понятийно-терминологическим аппаратом дисциплины (уместность употребления, аббревиатуры, толкование и т.д.), отсутствуют ошибки в употреблении терминов. Обучающийся самостоятельно, и отчасти при наводящих вопросах преподавателя, демонстрирует уверенное владение освоенным материалом, изложение сопровождает адекватными иллюстрациями (примерами). Ответ в достаточной степени структурирован, части ответа логически взаимосвязаны. Обучающийся способен анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал.
«3» (удовлетворительно, зачтено)		Содержание ответа в целом соответствует заданному вопросу. Обучающийся демонстрирует знание обязательного объема фактического материала по дисциплине, но оперирует неточными формулировками и допускает фактические ошибки (25–30%). Продемонстрировано владение понятийно-терминологическим аппаратом дисциплины, допущены ошибки в употреблении и трактовке терминов, расшифровке аббревиатур. Обучающийся проявляет затруднения в самостоятельных ответах. Примеры и иллюстрации, приведенные в ответе, в малой степени соответствуют изложенным теоретическим аспектам. Ответ плохо структурирован, части ответа разорваны логически. Обучающийся затрудняется анализировать, сравнивать, классифицировать, обобщать, конкретизировать и систематизировать изученный материал.
«2» (неудовлетворительно, не зачтено)		Содержание ответа не соответствует заданному вопросу или соответствует ему в очень малой степени Продемонстрировано крайне низкое (отрывочное) знание фактического материала, допущено много ошибок – практически все факты (данные) либо искажены, либо неверны.

		Продемонстрировано крайне слабое владение понятийно-терминологическим аппаратом дисциплины (неуместность употребления, неверные аббревиатуры, искаженное толкование и т.д.), присутствуют многочисленные ошибки в употреблении терминов. Обучающийся не освоил обязательного минимума знаний дисциплины, не способен ответить на вопросы даже при дополнительных наводящих вопросах преподавателя

4.5. Показатели и критерии оценивания курсовой работы

Оценка	Показатели оценивания	Критерии оценивания
«5» (отлично, зачтено)	Качество выполнения всех разделов курсовой работы; оформление, структура и стиль курсовой работы; самостоятельность выполнения, выполнение и сдача курсовой работы в установленные сроки.	Выполнены все разделы и задания курсовой работы; работы выполнена в срок; оформление, структура и стиль курсовой работы образцовые; курсовая работа выполнена самостоятельно, присутствуют собственные обобщения, заключения и выводы.
«4» (хорошо, зачтено):		Выполнены все разделы и задания курсовой работыс незначительными замечаниями; работа выполнена в срок; в оформлении, структуре и стиле работы нет грубых ошибок; работа выполнена самостоятельно.
«3» (удовлетворительно, зачтено)		Выполненные задания курсовой работы имеют значительные недочеты, устраненные после проверки преподавателем; работа выполнена с нарушениями графика; имеются недостатки по оформлению структуре и стилю работы; курсовая работа выполнена самостоятельно.
«2» (неудовлетворительно, не зачтено)		Часть работы выполнена из фрагментов работ других авторов и носит несамостоятельный характер; задания в курсовой работе решены не полностью или решены неверно; содержание работы не соответствует заданной теме; при выполнении работы не были использованы ключевые литературные источники; оформление работы не соответствует стандартным требованиям.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Для проверки качества освоения программы дисциплины и оценки результатов обучения по дисциплине, соотнесенных с установленными в программе индикаторами достижения компетенции проводится текущий контроль успеваемости и промежуточная аттестация обучающихся в форме экзамена, курсовой работы.

Контроль успеваемости обучающихся осуществляется с использованием рейтинговой системы оценки успеваемости обучающихся.

Текущий контроль проводится регулярно на всех видах групповых занятий по дисциплине. В конце семестра на основании поэтапного контроля процесса обучения суммируются баллы текущих, рубежных рейтингов (контрольные недели), подсчитываются дополнительные баллы (за посещаемость и активность на занятиях).

Результаты рейтинговой аттестации объявляются преподавателем на последнем занятии в зачетную неделю и служат основой для итогового результата промежуточной аттестации обучающегося по дисциплине.

5.1. Соответствие балльной шкалы оценок по дисциплине уровню сформированности компетенций обучающегося

Уровень сформированности компетенций	Оценка	Пояснение
Высокий	«5» (отлично) зачтено	Теоретическое содержание курса освоено полностью, без пробелов, все предусмотренные программой обучения учебные задания выполнены, планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций, достигнуты.
Выше среднего	«4» (хорошо) зачтено	Теоретическое содержание курса освоено полностью, все предусмотренные программой обучения учебные задания выполнены с незначительными замечаниями, планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций, достигнуты.
Средний	«3» (удовлетворительно) зачтено	Теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, большинство предусмотренных программой обучения учебных задач выполнено, но отмечены ошибки, планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций, в целом достигнуты.
Неудовлетворительный	«2» (не удовлетворительно) не зачтено	Теоретическое содержание курса не освоено, большинство предусмотренных программой обучения учебных заданий либо не выполнено, либо содержит грубые ошибки; дополнительная самостоятельная работа над материалом не приведет к какому-либо значимому повышению качества выполнения учебных заданий. Планируемые результаты обучения по дисциплине, соотнесенные с установленными в программе индикаторами достижения компетенций, не достигнуты.